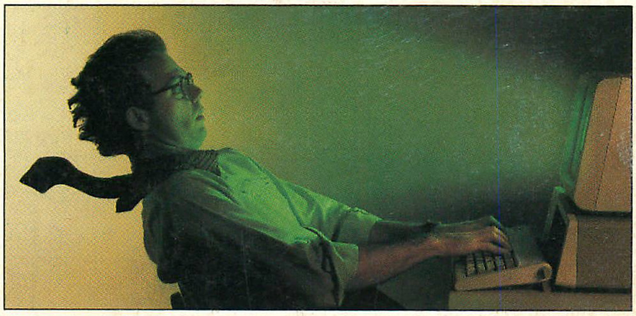
APRIL 1984 VOL. 1, NO. 7 $3.95

FOR IBM PERSONAL COMPUTER USERS

TECH JOURNAL

LIFE IN THE FAST LANE

In Search of a High-Resolution Timer



[redacted]

LISP FOR THE PC: THREE ALTERNATIVES

TALE OF TWO MICE: MICROSOFT AND MOUSE SYSTEMS

ASSEMBLY ROUTINES FOR MAKING SOUNDS ON THE PC

RESEARCH REPORT FOUR MAJOR COMPILERS AND THE ANS COBOL STANDARD

SORTING METHODS AND TIMING ON THE PC

0 14024 14357 04

THE WORLD'S BEST SELLING MULTIFUNCTION BOARD IS NOW EVEN BETTER

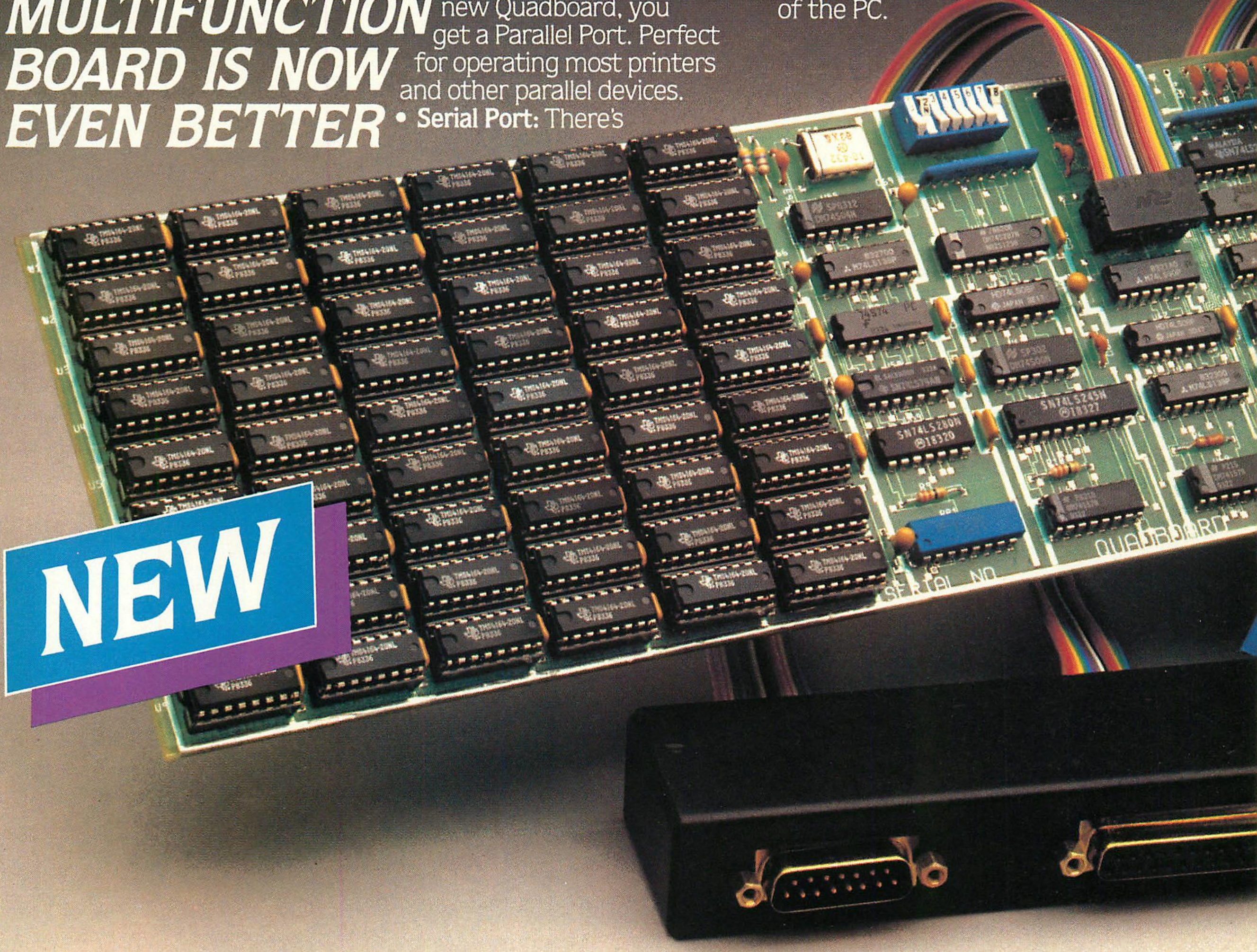
The new 384K Quadboard by Quadram is the most comprehensive board you can buy for the IBM PC or XT. Now with added hardware features and advanced software. But at a very low price.

NEW EXPANDED QUADBOARD
Quadboard now delivers 9 of the most needed PC functions/features. To let you get the most out of your Personal Computer. And help you work better and faster.

EXPANDABLE TO 384K

All of these features are standard on the new Quadboard:

- **Parallel Port:** With the new Quadboard, you get a Parallel Port. Perfect for operating most printers and other parallel devices.
- **Serial Port:** There's a Serial Port, too. Fully programmable, use it to connect to plotters, modems, and other serial devices.
- **Chronograph:** And Quadboard's Chronograph (Real-time clock/calendar) keeps your system's clock up-to-date.
- **Game Port:** The new Quadboard has an IBM compatible Game Port. Plug in a joystick or game paddles, and fire away.
- **I/O Bracket:** Quadboard now comes with a special I/O bracket. Use it to organize your expansion port connectors. Snaps right onto the back of the PC.



QUADBOARD®

- **Expandable to 384K:** The new Quadboard is expandable in 64K increments for up to 384K additional RAM. With full parity checking standard. With the new Quadboard and a fully populated system board, you can take your PC's memory up to the 640K limit.
- **QuadRAM Drive:** Plus, with Quadboard you get advanced QuadMaster Software. Including the QuadRAM Drive program. Use it to set up multiple RAM Drives in Quadboard memory. Solid state drives that let you store and retrieve data quickly and easily. Or take advantage of QuadMaster disk caching. To access frequently used data whenever you need it.
- **MasterSpool:** QuadMaster Software also includes MasterSpool. Use it to set up a software print buffer quickly and easily. This advanced spooler lets you pause at any time, back up or move forward in a file. Choose just the amount of buffer space you need and stop waiting on your printer.
- **Qswap:** Another feature of QuadMaster Software is Qswap. With Qswap change line printers 1 and 2 back and forth, with just a few keystrokes, as often as you like.

QUADBOARD STANDS OUT FROM THE PACK

Now more than ever Quadboard is the first and only board your IBM PC or XT may ever need. No other board even comes close. Because Quadboard is designed for performance. Engineered for dependability. And built in the continuing tradition of Quadram Quality.

QUADRAM QUALITY™

There are many imitators, but only one leader. So make sure you ask for Quadboard by Quadram, the leader in microcomputer enhancement products.

Compare. See why more Quadboards are bought than any other multifunction board...

Features/ Functions	Quadboard	SixPakPlus
Memory Available	0-384K	0-384K
Parallel & Serial Port	Yes	Yes
Clock/ Calendar	Yes	Yes
I/O Bracket	Standard	Optional
Game Port	Standard	Optional
Diagnostic Testing	Yes	Yes
Advanced Spooler	Yes	No
Simple Menu Setup	Yes	No
Disk Cache	Yes	No

SixPakPlus is a trademark of AST Research Inc.

QUADRAM CORPORATION™
An Intelligent Systems Company

4355 International Blvd./Norcross, Ga. 30093
(404) 923-6666/TWX 810-766-4915 (QUADRAM NCRS)

CIRCLE NO. 191 ON READER SERVICE CARD

BY QUADRAM™

Enhance Your IBM® PC With Two Exciting Quadram Products.

True or False: You Can Now Run Apple® Software in Your IBM® PC.

✔ True. Introducing Quadlink by Quadram. The revolutionary enhancement board that turns your IBM Personal Computer into an Apple-compatible system.

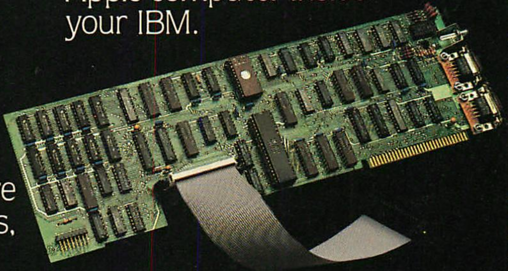
Quadlink. Simply plug it inside your PC, press a few keys, and instantly run most Apple software packages available.

Quadlink greatly enhances your PC's capabilities.

✔ True. Quadlink gives you access to the largest software library ever written. Business, educational, and entertainment packages. Software for any professional or home use. This means your PC can now do more than ever before. In more ways than ever before.

With Quadlink, there's no diskette reformatting needed.

✔ True again. With Quadlink you'll never have to worry about "compatibility." Just put your Apple diskette in the PC drive and watch it go. It's that easy. Like having an Apple computer inside your IBM.



QUADLINK™ BY QUADRAM™

And every Quadlink comes with that traditional Quadram Quality built right in.

✔ Absolutely true.

Quadlink is available at Quadram dealers worldwide.

✔ True. So visit the Quadram dealer nearest you today and ask to see Quadlink in action. And while you're there ask about our full line of IBM PC enhancements. Including the popular six-function Quadboard.™ You'll agree: when it comes to quality engineering and dependable performance, Quadram passes the test.

QUADRAM™ CORPORATION
An Intelligent Systems Company

4355 International Blvd./Norcross, Ga. 30093
(404) 923-6666/TWX 810-766-4915 (QUADRAM NCRS)

Computer stores were full of programs for the COMPAQ Portable before there *was* a COMPAQ Portable.

Here's why.

The COMPAQ Portable runs all the popular programs written for the IBM Personal Computer.

The COMPAQ™ Portable Computer runs more programs than any other portable computer. In fact, it runs more than most non-portable personal computers.

All of the popular programs written for the IBM® Personal Computer will run on the COMPAQ Portable. There are hundreds of them, and they run just the way they come out of the package.

Find out, "What if...?"

With the COMPAQ Portable and one of the popular spreadsheet programs, you can create a budget, a plan, a cost analysis or an estimate. In minutes you can do updates that normally take hours. You can examine several alternatives in less time than it would take to do a single problem by hand.

Spend more time thinking and less time crunching numbers.

Money management made portable

With the COMPAQ Portable you can take advantage of financial programs for professional money management.

Receivables programs tell you who owes money and since when. Payables programs give you control of outgoing cash. General ledger programs record financial histories. Payroll programs keep records and generate checks. Inventory management programs help you control investment.

You can take computerized accounting wherever you need it and keep closer contact with your company's financial status.

Word processing to go

Imagine the power of a *portable* word processor. Choose from a wide variety of programs to meet your specific writing application with the COMPAQ Portable.

With word processing programs and an optional printer you don't have to start fresh when you need to make revisions. You can change formats, move paragraphs, and automatically search for the parts that need to be updated.

In other words, you can work smarter—not harder.

Computer conversations

There are a number of communications expansion boards and programs available for the COMPAQ Portable.

Some allow you to access a variety of central computers over the phone. You can have headquarters information wherever you travel.

With others you can pull facts from data banks on thousands of topics.

Others enable you to send electronic mail.

A world of business information is available wherever you and the COMPAQ Portable go.

...and lots of others

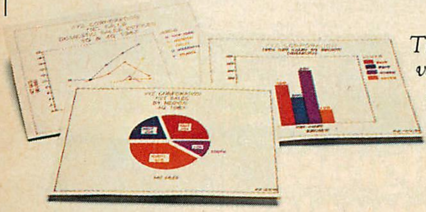
There are programs for designing graphs and charts, for tracking complicated schedules, and for electronic filing. There are powerful integrated programs that let you perform several tasks after entering data only once.

The wealth of programs and convenience of portability make the COMPAQ Portable the most useful personal computer on the market.

The COMPAQ Portable comes standard with one 320K byte diskette drive and 128K bytes of memory. A second diskette drive and additional memory are available.

The bottom line is this—you just can't buy a more practical, useful, productive computer.

For the location of the Authorized Dealer nearest you, call 1-800-231-0900.



The COMPAQ Portable runs a variety of programs for designing color charts and graphs that can enhance your business presentations.

COMPAQ™

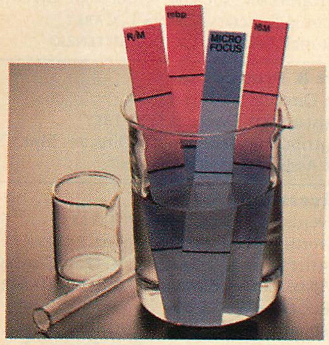
FOR IBM PERSONAL COMPUTER USERS
PC TECH JOURNAL
Volume 1 Number 7
April 1984

ARTICLES

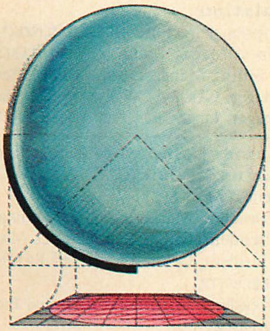
DEPARTMENTS

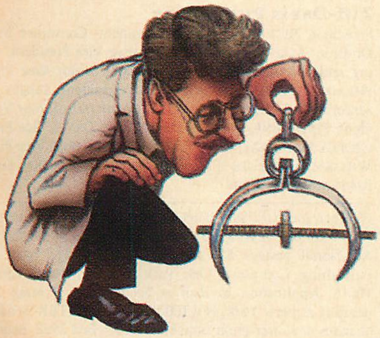
TECH NOTEBOOKS

PRODUCTS



76


106




150

PC TECH JOURNAL

VOL. 1, NO. 7

Publisher
Jeff Weiner

Editorial
Editor: Will Fastie
Managing Editor: Marjory Spraycar
Technical Editors: Julie Anderson, Susan Glinert-Cole
Copy Editor: Barbara Tilly
Contributing Editors: Ray Duncan, Richard Foard, Arthur A. Gleckler, Thomas V. Hoffmann, Sol Libes, Max Stul Oppenheimer
Editorial Secretary: Diana L. Carey
Editorial Assistant: Carole Autenzio

Art & Production
Art Director: Nancy Lepow
Assistant Art Director: Jane Frey
Contributing Assistant Art Director: Mary Shapiro
Art Assistant: Sandra Ray

Advertising Sales
Advertising Director: Newt Barrett
Advertising Coordinator: Sheryl Panzer
District Managers: Ian Smith—East Coast; Caroline Anderson—Midwest; Ted Bahr, Harriet Rogers—West Coast
Account Sales Manager: Pauline Scherer
Account Representatives: Rita Burke—East Coast; John Grogan—Midwest; Bill Bush, Arlene Steadman—West Coast

Circulation
Subscription Director: Chet Klimuszko, New Business Manager: Eric A. Bernhard, Renewal and Billing Manager: Shane Boel, Retail Sales Manager: Gerald Grossman, Telemarketing Manager: Linda Rossiter

Consumer Computers & Electronics Magazine Division
President: Larry Sporn
Vice President, Marketing: J. Scott Briggs
Vice President, Circulation: Carole Mandel
Vice President, General Manager: Eileen G. Markowitz
Vice President, Licensing and Special Projects: Jerry Schneider
Creative Director: Peter J. Blank
Editorial Director: Jonathan D. Lazarus
Promotion Manager: Ronni Sonnenberg

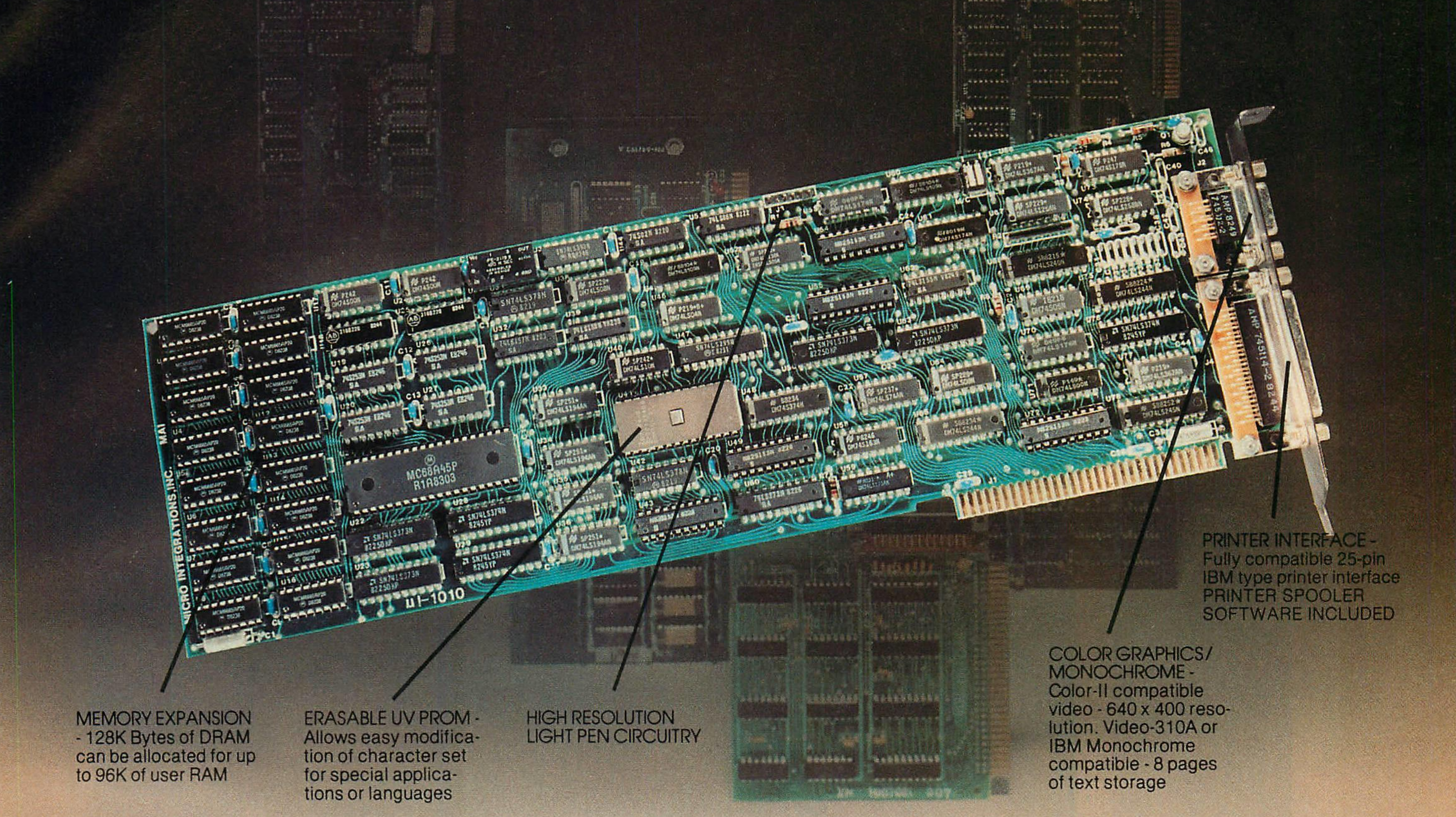
Ziff-Davis Publishing
President, Richard P. Friese; President, Consumer Magazine Division, Albert S. Traina; Executive Vice President, Marketing and Circulation, Paul H. Chook; Senior Vice President, Phillip T. Heffernan; Senior Vice President, Sidney Holtz; Senior Vice President, Edward D. Muhlfeld; Senior Vice President, Philip Sine; Vice President, Baird Davis; Vice President, George Morrissey; Vice President, Rory Parisi; Vice President, William L. Phillips; Treasurer, Selwyn Taubman; Secretary, Bertram A. Abrams

PC Tech Journal (ISSN 0738-0194) is published 12 times a year, $29.97 for one year, $52.97 for two years, $69.97 for three years. Additional postage $12 for Canada & Foreign by Ziff-Davis Publishing Company, One Park Avenue, New York, N.Y. 10016. Application to mail at Second-Class Postage Rates is pending at New York, NY 10001. POSTMASTER: Send address changes or subscription inquiries to P.O. Box 2968, Boulder, CO 80321.

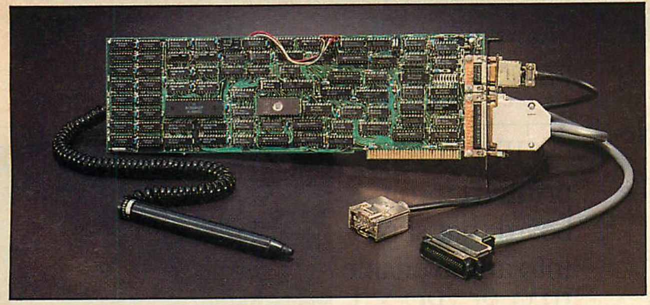
Business Offices: Advertising, *PC TECH JOURNAL*, One Park Avenue, New York, NY 10016. 212-725-7947. Subscription inquiries to *PC TECH JOURNAL*, P.O. Box 2968, Boulder, CO 80321.

Editorial Office: *PC TECH JOURNAL*, The World Trade Center, Suite 211, Baltimore, MD 21202. 301-576-0770. The Source ID STY682.

Put graphics & multi function in one slot.



This IBM/PC compatible board provides memory expansion (128 KBytes of DRAM), monochrome or color video, light pen, and printer interface on a single plug-in board. The MAI board's high-density memory (up to 8 pictures storage) provides unexcelled color graphics enhanced up to 640(H) x 400(V) resolution in a wide variety of operating modes. Modes are selectable right on the board.



Now available with Halo* software for 320 x 200 16-color, 640 x 200 4-color, and 400 vertical line resolution modes. (*Halo is a registered trademark of Media Cybernetics, Inc.)

COMPARISON CHART

	COLOR		MONOCHROME	
	IBM	AMDEK	IBM	AMDEK
SLOT 1	Color Graphics Board	MAI Board	Black & White/ Parallel Board	MAI Board
SLOT 2	Printer Board			
SLOT 3	RAM Board 1 Slot Left*	3 Slots Left*	RAM Board 2 Slots Left*	3 Slots Left*

*When Using Disk Controller

Just circle the reader service number, or call direct for complete technical specifications and prices.

2201 Lively Blvd. • Elk Grove Village, IL 60007
(312) 364-1180 TLX: 25-4786

AMDEK

REGIONAL OFFICES: Calif. (714) 662-3949 • Texas (817) 498-2334

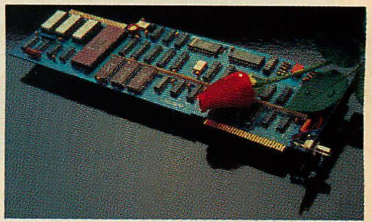
CIRCLE NO. 230 ON READER SERVICE CARD

Amdek . . . your guide to innovative computing!

She's Director of Decision Support, the leader in IBM 3270 mainframe/PC networks.

IRMA is taking charge! Her performance in over 25,000 IBM 3270 mainframe/PC installations has made her the industry's leading Decision Support Interface.™ Why? Because IRMA works. Easily and economically, IRMA's on the job giving PC users mainframe data access, selection, storage and communication back to the mainframe.

She's more capable than ever.



IRMA is a printed circuit board that slips into any available IBM PC/PC XT slot and attaches via coaxial cable to a 3270 controller. She's as easy to work with as the PC itself and provides standard text file transfer software

IBM PC and IBM PC XT are trademarks of International Business Machines Corporation. Apple Lisa is a trademark of Apple Computer, Inc.
DEC Rainbow is a trademark of Digital Equipment Corporation.

for VM/CMS and MVS/TSO mainframe environments, in addition to optional APL Terminal Emulation.

She's got more to offer.

Other IRMA products give you more ways to build cost-effective IBM 3270 and IBM PC-based decision support networks.

IRMALINE™ is the first coaxial link between 3270 networks and remote personal computers, including the IBM PC/PC XT, the Apple Lisa, the DEC Rainbow and others.

IRMALETTE,™ the Asynchronous Interface, lets you run the same IRMA software in your remote personal computer that you run in the office.

So connect with success. With IRMA products from DCA. Call 1-800-241-IRMA.

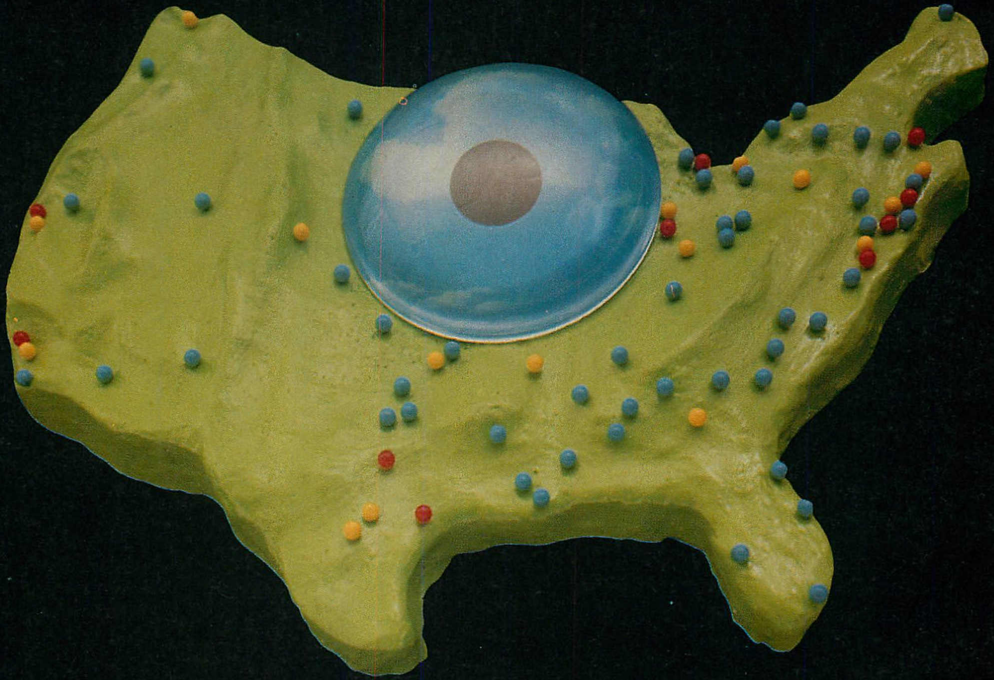
Connect with success.

dca®

Digital Communications Associates, Inc.

CIRCLE NO. 210 ON READER SERVICE CARD

Our Vision Is Peripheral



Yours Can Be, Too, With Multi-Tech Systems' New Modem For The IBM PC

Expand your horizons. Connect your IBM PC to San Diego, San Antonio, Savannah, and points between. Link up to New York for the Dow Jones News, or check out The Source in D.C. Dial up CompuServe in Ohio, or the corporate mainframe downtown. Swap programs with your pal in Seattle. Transfer files with your friend next door.

Share the latest excitement at Multi-Tech: the MT212PC Intelligent Modem for the IBM PC.

- Plugs into any IBM PC expansion slot.
- Transmits at 0-300 or 1200 bits per second.

MT212PC Board Modem

- Automatically dials, answers, hangs up, redials.
- Stores phone numbers, parameters, log-on sequences.
- Creates, stores, prints, displays, and transfers files.
- No IBM serial card needed . . . price includes all software.

At Multi-Tech Systems, our future is new markets for us, and new products for you. Since 1971, we've been building acoustic couplers, direct-connect modems, testers, multiplexers, and now this newest generation of intelligent modems.

The MT212PC modem has a two-year warranty and a suggested retail price, including companion software, of just $549. For more information, write to Multi-Tech Systems, Inc., 82 Second Avenue SE, New Brighton, Minnesota 55112. Or phone us at 612-631-3550.

TWX 910-563-3610

MULTI-TECH SYSTEMS
The Peripheral Vision People

Trademarks: IBM Personal Computer (IBM PC): International Business Machines Corporation; Dow Jones News/Retrieval Service: Dow Jones & Company, Inc.; The Source: Source Telecomputing Corporation, a subsidiary of The Reader's Digest Association, Inc.; CompuServe: CompuServe Information Services Division, Columbus, OH.

CIRCLE NO. 204 ON READER SERVICE CARD

Textra™ word processing

The shortest distance between your thoughts and the printed word.

Your thoughts come faster than your words. Capturing them smoothly and quickly is the strength of Textra, and the secret behind its success.

Textra blends advanced features with a simplified design that makes it easier to write with your personal computer. And that's what word processing is all about.

Start with the 'on-line' tutorial, which has been called everything from elegant to excellent. It's the finest introduction to a software program ever filmed. You'll be ready to write in record time.

And Textra will be ready for you. Its highly responsive full screen editor helps you put your thoughts on the screen swiftly. Automatic reformatting and on-screen bolding and underlining show you what your printed text will look like at all times. You can even preview your pages *before* they're printed, so you only have to print them once!

Whether you're searching for your first word processor, or feel miserable with the one you have now, take a close look at Textra. Ask your dealer for a demonstration today.

It's guaranteed to open your eye.

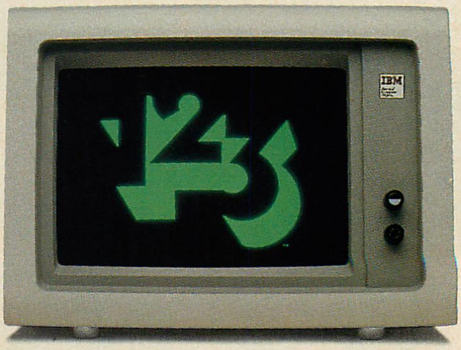
Textra runs on the IBM PC and compatibles, and requires PCDOS (any version), 128K, and either monitor. Textra Jr. ($39.95) requires 64K.

Ann Arbor Software

407 N. Main, Ann Arbor, MI 48104 Phone (313) 769-9088

CIRCLE NO. 228 ON READER SERVICE CARD

HERCULES™ is the only graphics card that can run 1-2-3™ on IBM's® monochrome display.



And that's just for starters.

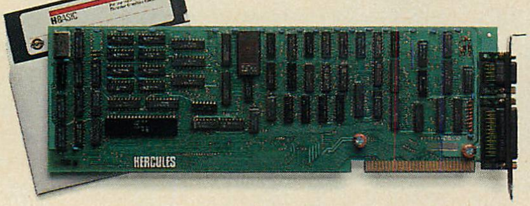
You don't have a Hercules Graphics Card? Then unfortunately you won't be able to run 1-2-3, including all its graphics, on IBM's monochrome display. And you'll miss out on all the other reasons why there are more Hercules Graphics Cards producing more high resolution graphics than any other add-on card for the IBM PC.

But don't take just our word for it. If you need convincing, remember that most of the IBM PCs at Lotus™ are running Hercules Graphics Cards. And the authors of 1-2-3 know a good card when they see one. Or consider that the Hercules Graphics Card is widely used at Rockwell, Mass Mutual, and Carnegie Mellon. They couldn't all be wrong, could they?

At $499, we think the Hercules Graphics Card offers the best price/performance ratio of any graphics card available today. As you can tell, plenty of users agree with us.

Call or write for our free information kit. You'll see why the first graphics card for the IBM PC is still the best.

Hercules Computer Technology, 2550 Ninth St., Suite 210, Berkeley, California 94710. Telephone: (415) 540-6000.



Hercules. We're strong on graphics.

CIRCLE NO. 151 ON READER SERVICE CARD

The Hercules Graphics Card offers 720×348 graphics resolution on IBM's monochrome display, compatibility with text mode software, a parallel printer port, software to use BASIC's graphics and a two year warranty. A graphics subroutine library with screen dump is available separately for $50. Graphics software that is compatible with the IBM color graphics card but does not have a Hercules compatible version will not run on the Hercules Graphics Card. **Foreign Distributors:** Reflex/U.K.; Computer 2000/W Germany; Edisoft/France. Hercules Graphics Card is a trademark of Hercules Computer Technology. IBM is a registered trademark of International Business Machines. 1-2-3 and Lotus are trademarks of Lotus Development.

TECH JOURNAL
VOL. 1 NO. 7

PC/IX

UNIX Comes to the PC with a Bang . . . and a Whimper

WILL FASTIE

BANG! IBM introduces PC/IX, a version of UNIX, for the PC.

Whimper . . . The system is single-user, costs $900, requires an XT, occupies 7.5 megabytes (19 floppies), will be available directly from IBM only through National Accounts (NAD) and National Marketing (NMD) and not through all regular PC distribution channels, and is the only UNIX-related announcement IBM made (that is, there are as yet no IBM-supplied applications for the new environment).

THE BANG

IBM's announcement was explosive for two reasons: IBM and Microsoft.

To begin with, IBM has officially endorsed UNIX for the first time, and in a big way. It is true that the IBM 9000, IBM Instrument's 68000-based system, is already available with Microsoft's XENIX, but the 9000 is a niche product designed to compete in the laboratory, scientific, and academic markets in which UNIX is already somewhat established, especially on minicomputers. The PC obviously has wider market appeal, particularly in the Fortune 2000 companies that make up IBM's major market thrust. UNIX is suddenly available to that market from the original equipment vendor, and that's significant. Of course, IBM's announcement of PC/IX makes UNIX a de facto standard by, in effect, acknowledging the long-standing efforts of the UNIX software industry. And, of course, IBM has bowed (or at least tipped its hat) to AT&T, whose Bell Labs originally developed the system. IBM even pays the royalty: Interactive Systems Corporation, of Santa Monica, California, developed PC/IX from its own product—IN/ix (formerly IS/3)—and is a licensee of AT&T Technologies, Inc.

As for Microsoft, MS-DOS (and therefore PC-DOS) has always been touted as the bridge to XENIX. DOS 2.0 has a number of UNIX-like commands, including a few that are even spelled the same as in UNIX, and the command syntaxes of the two operating systems are very similar. MS-DOS 3.0 is rumored to carry this resemblance even further. So when XENIX was chosen for the IBM 9000, Microsoft was seen to have the inside track. What happened?

We may never know the answer. Two possibilities spring to mind, both having to do with Bill Gates. After successfully bringing Microsoft to the forefront of the IBM PC world, Gates turned his attention first to Tandy and the Model 1000 and then to Apple and Macintosh. To his credit, both of these projects, like the original IBM PC development, are innovative and important. However, they might have affected the resources dedicated to XENIX development. It is even possible that IBM considered the Apple connection a little too sticky.

What an impact. UNIX has suddenly become a standard, Microsoft is suddenly not a spotlighted UNIX player, and the UNIX industry is suddenly on a roll.

THE WHIMPER

I know what you're thinking. "Will, how the heck can you look at IBM's big bang and then turn around to call it a whimper?" I gave most of the reasons above, and I think they are self-explanatory. A few more comments, however, may be needed.

First, $900 is not a bad price for PC/IX. As near as we can tell, PC/IX has just about everything a software developer might expect in a UNIX implementation, including MAKE and SCCS. There is even a sophisticated screen editor called INed that apparently replaces VI. Of course, the standard UNIX repertoire of C, tools for C, the assembler, special processors (M4, YACC, LEX), text formatting tools, and even a SNOBOL III interpreter is provided. Also included are four binders of documentation and extensive on-line help. There is even one diskette full of games!

That's a lot for the money. However, it is all bundled together, which makes development of saleable applications an expensive proposition, because the end-user must purchase the whole $900 package in order to run the first such application. It goes without saying that few inexpensive applications will emerge

AT LAST! TEN IBM PC/XT FUNCTIONS AVAILABLE THRU ONE SLOT!

MAGNUM HARDWARE AND SOFTWARE FEATURES:

1. **SuperSmart MODEM** - 212A type (300-1200) intelligent modem with Auto Answer/Auto Dial (touchtone/pulse). Mounts inside PC/XT where it belongs. 52 number Auto-Dialer optional.
2. **PARALLEL PRINTER PORT**
3. **SERIAL PRINTER/COMM. PORT.**
4. **GAME/MOUSE PORT**
5. **QUARTZ CLOCK/CALENDAR**
6. **MEMORY** - 64K to 512K bytes
7. **TurboRAM** - Extra RAM used as hard disk.
8. **TurboSpool** - Print job buffer/spooler.
9. **TurboComm** - SuperSmart MODEM control program includes: Single key, built-in emulation of such popular terminals as DEC VT-100 (ANSI standard), Data General 605X series, ACT-5A and ADM-5A. Supports all standard baud rates to 19,200. Single keystroke connection to host. Screen transmission status reporting. Full printer and disc drive controls. Full buffering and interrupt driven I/O). Error detection and logging. Capability to interactively download and upload disk files.
10. **TurboFile** - TurboComm's file transfer utility featuring: Automatic unattended file transfer initiated any time, day or night. Transfers any file supported by PC/DOS. Automatic error detection and recovery. Password computer access protection.

CHECK OUR STANDARD FEATURES AND LOW COST OPTIONS

	BASE PRICE	SUPERSMART MODEM	PARALLEL PORT	SERIAL PORT	MOUSE/ GAME PORT	CALENDAR/ CLOCK	64K BYTES MEMORY	AUTO DIALER	TURBO-COMM	TURBO-FILE	TURBO-RAM	TURBO-SPOOL
MAGNUM MODEM	595.	✓				✓	325.	75.	✓	✓	50.	50.
MAGNUM MEMORY	425.						✓				✓	✓
MINI-MAGNUM	695.	✓	✓	✓	✓	✓	325.	75.	✓	✓	50.	50.
MAGNUM 10	1195.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

ADD 75.00 PER EACH ADDITIONAL 64K OF MEMORY UP TO 512K BYTES

Order now! Check, M.O. Visa and Mastercard accepted • Satisfaction guaranteed. 2 years warranty

MULTI-FUNCTION AT IT'S BEST FROM

AMERICAN HIGH-TECH INDUSTRIES, Inc.
ELEVEN-ONE-ELEVEN WILCREST GREEN DRIVE
SUITE 201, HOUSTON, TEXAS
(713) 952-7332

AMERICAN HIGH-TECH

CIRCLE NO. 104 ON READER SERVICE CARD

Fill in your IBM micro/mainframe communications picture.

AST Research, the leader in IBM PC enhancement products, brightens your micro/mainframe communications picture with a full palette of economical, integrated hardware/software masterpieces. With AST Products, you can emulate IBM terminals or create PC-based Local Area Networks.

AST improves your office operating cost picture.

AST communications products give your IBM PC the flexibility to act as a terminal for your host system or as a stand-alone computer for smaller tasks. Your PC won't bog down the mainframe with unnecessary small jobs and local computing on the PC eliminates phone line charges too. Get the power of a mainframe when you need it and personal computer convenience right at your fingertips.

Applications solutions that are strokes of genius.

AST keeps pace with your ever-changing applications requirements with reliable, high quality, cost effective communications products. AST products provide support for Bisync and SNA/SDLC communications protocols as well as networking multiple PC's for sharing resources.

Choose AST products — by the numbers.

These AST communications packages are currently available:

1. **AST-SNA™** emulates a 3274/3276 controller and 3278 or 3279 display terminal using SDLC protocol.
2. **AST-BSC™** emulates 2770 batch RJE and remote 3270 display terminals using 3270 Bisync protocol.
3. **AST-PCOX™** allows your PC to connect to an IBM 3274/3276 cluster controller via coax cable and emulates a 3278 or 3279 display terminal.
4. **AST-3780™** emulates 2770, 2780, 3741, and 3780 RJE workstations using Bisync protocol.
5. **AST-5251™** emulates a 5251 Model 12 remote workstation connected to an IBM System 34, 36 or 38.
6. **PCnet™** is the first Local Area Network designed specifically for the IBM PC or XT and the PC-DOS 1.1 or 2.0 operating system.
7. **CC-232™** is a user-programmable dual-port card capable of communicating in Async, Bisync, SDLC, or HDLC protocols.

CIRCLE NO. 109 ON READER SERVICE CARD

Discover how well AST can fill in your micro/mainframe communications picture. For descriptive data sheets, write or call: AST Research Inc., 2121 Alton Ave., Irvine, CA 92714. (714) 863-1333. TWX: 295370ASTRUR

AST RESEARCH INC.

Number One Add-Ons For IBM PC.

IBM is a registered trademark of International Business Machines Corporation. PCnet is a registered trademark of Orchid Technology, Inc. AST-3780 is a product developed by AST Research, Inc., and Wilmot Systems, Inc. AST-5251 is a product developed by AST Research, Inc., and Software Systems, Inc., of Jefferson City, MO. PCOX is a product developed by CXI Inc.

PC Owners . . .

Reach for Your Phone! This Winchester is Loaded . . . with UNIX Software.

That's right, partner. Now is the time to upgrade your PC with the *Sundown*™ disk. Includes controller. Installs right inside your PC in less than 10 minutes. Backed by our full one-year warrranty.

But that's only half the story . . . The *Sundown* comes loaded with *VenturCom Venix/86*. This highly-acclaimed operating system is a licensed implementation of AT&T's UNIX and is the only MULTI-USER, MULTI-TASKING UNIX environment available on the IBM PC. Plus you can store and run your MS/DOS programs and files as well!

We offer immediate delivery. And our price . . . now that will blow your boots off! Need we say more? Reach for your phone and dial:

617-491-1264

Unisource Software Corp., Department 4103
71 Bent Street, Cambridge, MA 02141

*UNIX is a trademark of Bell Laboratories.

CIRCLE NO. 135 ON READER SERVICE CARD

for PC/IX, in striking contrast to the large library of PC- and MS-DOS programs now available.

The absence of applications is, in fact, the biggest drawback. The product is clearly aimed at the system developer and, more specifically, the Fortune 2000 system developer. This market segment has the money to buy not only multiple licenses for PC/IX but also the expensive applications to go with it. More importantly, they have the resources to build their own in-house systems, just as they have been building systems for IBM mainframes in the past. UNIX might be a good choice in such cases; distributed computing and inter-computer communications are complex systems problems requiring sophisticated operating environments.

In short, the new product is not going to have mass market appeal for the moment. Expect the other UNIX players to have more aggressively priced systems of the unbundled variety available soon.

New Players

IBM chose to announce PC/IX just before the January UniForum conference and show in Washington, D.C., and the product was demonstrated there by representatives of IBM and Interactive Systems. The reaction of others in the UNIX business to the IBM announcement was very favorable. In fact, other vendors at the show seemed to be gleeful. There was not a frown to be seen, and the show was very busy and very, very crowded. (In deference to the growing importance of UNIX, the attire of the attendees was more dressy than in previous years.)

The show also managed to be a turning point for several firms, who will now become the new players in the continually unfolding IBM drama. First and foremost, there is Interactive Systems, a company well known in the UNIX community but, until now, little known elsewhere. INTERACTIVE employs about 150 people and has revenues exceeding $10 million, although the exact figure for the privately held firm is not available. President Tom Cull refers to Interactive, in business since 1977, as the oldest and largest independent UNIX software company. Other than PC/IX, Interactive's newest product is TEN/PLUS, a friendly front end to UNIX that presents a simple, consistent command set and a visual interface to the user. Many TEN/PLUS concepts are embodied in the PC/IX product, according to Cull.

The other new player is actually an old player, and part of an interesting "magic triangle" of companies. Digital Research (DRI) recently announced that it has agreed to join with Intel to prepare a version of UNIX System V for the 80286 processor. Previously, Intel and Microsoft were developing UNIX for the 286, an arrangement that has been dropped in favor of the DRI UNIX deal; Microsoft indicated that it wished to devote full energies to its own XENIX-286 project.

Intel also has an agreement with AT&T Technologies to develop UNIX; apparently the resultant product will belong to AT&T but can be marketed by both DRI and Intel on a non-exclusive basis. Furthermore, at UniForum DRI and AT&T announced an agreement to jointly develop a UNIX-based library of applications that both companies could sell; the applications would be designed for System V.

Intel's interest in such an effort for the 286 is clear. AT&T has a vested interest in any UNIX development, especially one relating to the recently announced System V. What is most interesting, though, is the DRI connection. After almost three years of taking it on the can for the miss of CP/M on the PC, DRI is clearly back in the middle of things. If their new agreements work out, DRI could very well become the Microsoft of the UNIX world, although the company might not use quite that description.

What's It All About, Really?

For all the excitement, PC/IX is only the tip of the iceberg. The product is a single-user, multi-tasking system, which is something of a strain for the PC as it is. Inexpensive multi-user systems are an obvious objective for the customer base, but the PC is probably not up to it.

I think IBM is giving us fair notice. A machine based on the Intel 80286 is widely rumored and seems to make sense. The 286 is powerful enough to run a multi-user UNIX environment and to support multiple terminals. But if such a machine existed today, the dearth of applications software would make marketing difficult at best. By making UNIX available now, IBM is indicating a future path and thus giving software developers a chance to respond with applications, products that can quickly port to the more powerful IBM PC of the future.

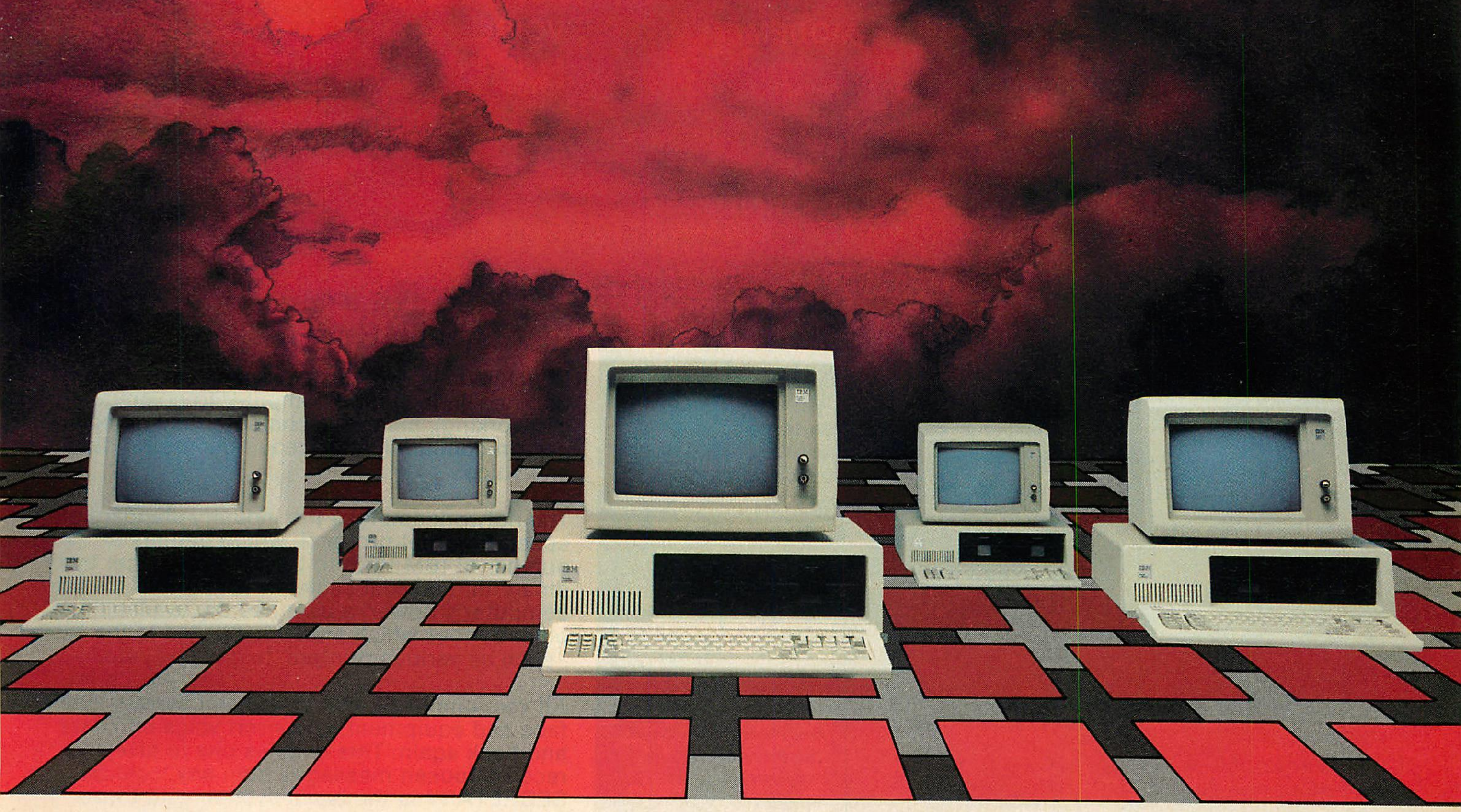
Information about PC/IX is available from IBM's National Accounts and National Marketing Divisions. Representatives from both divisions are usually available through any IBM branch office. Further information can be obtained from:

IBM Corporation
National Accounts Division
1133 Westchester Avenue
White Plains, NY 10604
914-696-1900
CIRCLE 491 ON READER SERVICE CARD

IBM Corporation
National Marketing Division
4111 Northside Parkway
Atlanta, GA 30327
404-238-2000
CIRCLE 492 ON READER SERVICE CARD

Interactive Systems Corporation
1212 Seventh Street
Santa Monica, CA 90401
213-450-8363
CIRCLE 493 ON READER SERVICE CARD

X-NET™ Local Area Network Broadens The Horizons Of All IBM PC's.*



Breakthrough! X-NET and the industry's leading accounting package have been linked to broaden the capabilities of all IBM PC's.

Here are the benefits of this breakthrough and what they mean to you.

1. READY ACCESS TO ALL INFORMATION. With X-NET, any computer can get information from any other computer on the network. This means more than one operator can be entering orders, invoicing, changing inventory... all at the same time! But where security is needed, you can keep any file private while still within the network.

2. YOU CAN START SMALL... Even if you have as few as two PC's, you'll have a system that is easily expandable and will grow with your company.

3. ...OR GET BIGGER. X-NET grows as you grow, and networks up to 255 PC's, and best of all, expansion is so simple, you can do it yourself.

4. SUPERIOR ACCOUNTING SOFTWARE. X-NET's software includes Accounts Receivable, General Ledger, Accounts Payable, Order Entry, Inventory Control and Payroll. You can still use your Spread Sheet and Word Processing software, but now much more effectively because your computer power is tied together.

5. ELECTRONIC MAIL. Send, receive and record inter-office communications without ever leaving your desk or interrupting an important business meeting.

6. AFFORDABILITY. X-NET is truly affordable to any company doing computing on any level. And the software is priced comparably to what others are charging for no-growth single-user software.

7. NO COSTLY CENTRAL FILE SERVER. X-NET eliminates the need of having an expensive computer to act as a Central File Server, something other networks require.

8. SHARE YOUR PERIPHERALS. No longer does each of your computers need its own set of peripherals. Any computer on X-NET is linked to your printers, plotters, disk-drives, etc. A huge money-saver!

Get the best of both worlds. The X-NET Local Area Network with superior software packages, and XCOMP's other hardware accessories. Call your local dealer or XCOMP today.

***X-NET also operates on IBM PC compatibles.**

XCOMP

XCOMP, Inc.
3554 Ruffin Road South
San Diego, CA 92123-2502
619-573-0077
TELEX: 182786

CIRCLE NO. 108 ON READER SERVICE CARD

ELAN™* Extended Local

NETWORKING
- Ethernet
- Electronic mail
- Device sharing
- No dedicated server required
- Floppys not required in single node
- Remote job execution
- File lock out
- Password protection

TELEPHONE SUPPORT
available through modem board)
- Autodial - with extended telephone numbers
- Call Forwarding
- Length Of Call Indicator
- Directory Support - with unlimited number storage
- Message Support - either leave or obtain messages. Receive messages remotely
- The modem board, with a handset, substitutes for a telephone

MODEM
- Gateway to other networks
- Electronic mail
- Device sharing
- Remote job execution
- File lock out
- Password protection
- 300 to 1200 baud modems available
- Data in network can be obtained by decoding touch-tone sequence or through voice recognition prompt
- Respond to remote terminal
- Access dictating systems on network. Control them by touch-tone decoding
- Programs or calculator can be used remotely. The touch-tone keypad can be used to provide numeric input to programs or the calculator from a remote site (programs are loaded by decoding touch-tone sequences).

VOICE
- Voice mail
- Voice annotated text
- Voice messages

VO!CE RECOGNITION
- Transparent keyboard. Speak instead of type

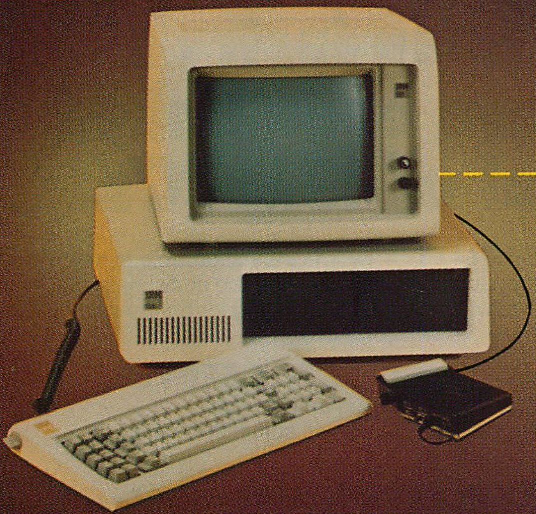
ELAN: THE MOST POWERFUL

ELAN is designed to meet your total communication needs, including computer to computer (networking), person-to-computer and person-to-person communication requirements for data and voice.

The network employs the industry-standard high speed Ethernet protocol, which permits a number of IBM PC's to be linked together by ordinary thin coaxial cable. In addition to his own computer's power, a user has the availability of other devices which are also attached to the cable - such as various printers, large disks, etc.

All versions of ELAN include an Ethernet interface and equipment to convert voice into data and back again. This enables one to give and receive spoken messages from any location. The SECRETARY is the basic system with these features.

SECRETARY $1695[1]
- Ethernet Link
- Ethernet Companion
- ELAN Software



EXECUTIVE $2995[1]
- Ethernet Link
- Ethernet Companion
- Modem (300 Baud)[2]
- Voice Recognition
- Microphone
- ELAN Software

[1]With 1st MATE, 2nd MATE, or 3rd MATE in Station
[2]Option: 1200 Baud Modem
*ELAN (formerly ComNet)

Area Network from TECMAR

THE POWER BEHIND THE PC

DICTATING SYSTEM

- Control the Pearlcorder X-02 or XR dictating system from keyboard, footpedal, or remote telephone keypad. Dictate to the network from anywhere in the U.S. or Canada.

FOOT PEDAL SUPPORT

- Controls dictating system
- Controls response to voice recognition

SOFTWARE

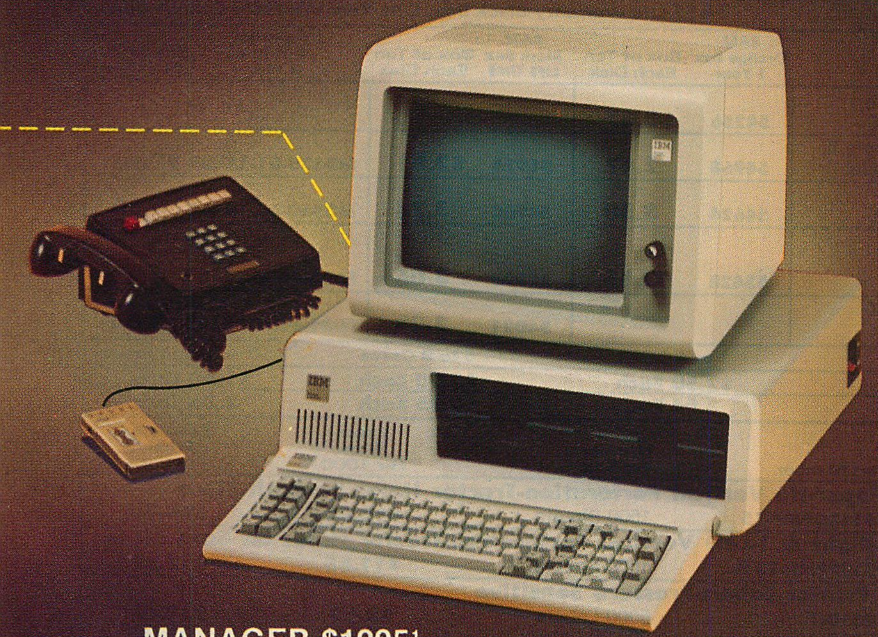
- Time Management - alerts you to appointments at any station you are logged onto
- Message Management - either electronic mail or voice
- Clock/Calendar - either visual or audible
- Calculator - either visual or audible
- Voice Management - oversees voice mail, voice message and voice annotated text operations

These software packages can be operated through voice recognition with voice output, through the telephone keypad with voice output or through the IBM PC keyboard

NETWORK FOR THE IBM PC

The MANAGER system adds a modem which can turn the PC into a telephone if a separate handset is added. The modem enables the MANAGER to receive unattended voice and data from any telephone in the U.S. or Canada. The MANAGER can accept commands by decoding the tones in the telephone keypad.

The EXECUTIVE is the most complete implementation of ELAN, adding computer recognition of spoken commands. An executive might phone the PC to leave or retrieve messages or request specific information. The PC, in a spoken voice, can request the user's access code or offer a list of options the user can select. The EXECUTIVE can then key in answers or commands with the phone's tone dialing buttons, or may simply speak his response to the computer.



MANAGER $1995[1]

- Ethernet Link
- Ethernet Companion
- Modem (300 Baud)[2]
- ELAN Software

HARDWARE

ETHERNET LINK

Permits communications between computers at extremely high speeds (10 Mbits per second). The transmission mode is through single video coaxial cable with easy-to-use BNC connectors.

ETHERNET COMPANION

Performs the function of voice digitization and voice replay, dictation machine control and foot pedal control. Also contains interface for mouse.

MODEM

103 (300 Baud) $295
212A (1200 Baud) $695

- 103 (300 Baud) or 212A (300 or 1200 Baud)
- Pulse/tone automatic dialer
- Dual tone DTMF receiver (decodes touch tones)
- Auxiliary voice circuit
- Auxiliary, optically coupled, ring indicator output (capable of being used for auto power-on)
- Can replace telephone with the addition of a handset

VOICE RECOGNITION $995
MICROPHONE $170

User-dependent 100 word recognition (200 words optional) with 98% accuracy. Permits computer to respond to voice input.

CIRCLE NO. 154 ON READER SERVICE CARD

TECMAR

Tecmar, Inc.
6225 Cochran Road
Cleveland, Ohio 44139
Phone: 349-0600 Telex: 466692

WHEN YOU NEED A DEPENDABLE DISKETTE, LINE PRINTER PAPER & LABELS DON'T SETTLE FOR LESS THAN THE BEST!

As low as $2.47 each

VERBATIM 5 Year Warranty — **BASF** Life Warranty-Black Box, One Year Warranty-Orange Box

Description	Compatibility	BASF Orange Box 1 Year	Box of Ten Each Disk	BASF Black Box LIFE TIME	Box of Ten Each Disk	Verbatim Soft Box 5 Year	Box of Ten Each Disk
5-1/4" single-side, single-density, soft sector	Apple II, Commodore, Radio Shack TRS-80	54256	$2.47				
5-1/4" single-side, double-density, soft sector	IBM PC (single-side), Apple II, Commodore, TRS-80 Model III	54968	2.47	54974	$2.81	MD525-01-18158	$2.66
5-1/4" double-side, double-density, soft sector	IBM PC (dual side)	54624	3.50	54980	3.74	MD550-01-18188	4.10
8" single-side, single-density, soft sector IBM Format 2305830	Radio Shack TRS-80 Model I	53428	2.55	54998	3.06	FD34-1000-3718	2.66
8" single-side, double-density, soft sector	Radio Shack TRS-80 Model II			54021	3.23	FD34-8000-18137	3.40

Description	Part Number	1 to 4 Cases	5 to 9 Cases	10 or More Cases
5-1/4" Library Case	Mini-K/10	$2.50 Each	$2.00 Each	$1.50 Each
8" Library Case	KAS/10	3.94 Each	3.15 Each	2.36 Each

Cases sold only with another item.

Description	Part Number	Number Per Box	Exact Price Per 1000	Estimate Price Per Box
14-7/8" x 11", 132-Column Fanfold Paper-No Vertical Perforation-Tractor Holes on Both Sides				
One Part 1/2" Green Bar	14112GBMP-15	1500 Sheets	$16.04	$24.06
9-1/2" x 11", 80-Column Fanfold Paper, 1-Right & 1-Left Vertical Perforation-Tractor Holes on Both Sides				
One Part Blank	9510PPMP-15 lb.	1500 Sheets	$11.68	$17.52
Nashua Line Printer Labels-Pressure Sensitive				
3-1/2x15/16 White One Up-One Wide	10350-1	5000 Labels	$2.60	$13.00
3-1/2x15/16 White Four Up-Four Wide	10350-4	20,000 Labels	2.60	52.00

Shipping Charges & Taxes Will Be Added To Above Prices

TELEPHONE AT NO CHARGE-FREE DIRECT TO OUR ORDER DESK

Use Your MASTER CARD; VISA/BANK-AMERICARD; AMERICAN EXPRESS CARD

From Nebraska *Phone* 800-642-9333 From Any Other State *Phone* 800-228-9288

DIGITAL SUPPLY 1221 HARNEY St. OMAHA, NEBRASKA 68102

CIRCLE NO. 271 ON READER SERVICE CARD

Letters to the Editor

APL for the PC

Bob Smith's review, "Two APL Systems," in your Nov-Dec issue was outstanding. Rarely have I seen such meticulous care and attention given to such a subject. Yet it did not become boring. Quite an achievement!

One problem with IBM's APL for the PC is that the color monitor and its associated color card have to be used. The monochrome monitor is superior in clarity and easier on the eyes, but up to now it was not possible to use it. I have found that the APL character generator chip from Computronix (1638 Fairgreen Dr., Fullerton, CA 92633) is excellent for just this application. It is easy to install and gives results far superior to the color monitor.

Your journal is excellent—the best in the field, I think. Keep up the good work.

John Collins
Laguna Beach, CA 92651

Editors, Again

I would like to echo the sentiments of Mr. Marsland on the Personal Editor (letter, Jan 1984) with one caveat. The editor does not have a true search/replace function, which seriously limits its usefulness as a real editor. In fact, all I want in the way of an editor is the Personal Editor with a true search/replace function. I am writing this letter partly in the hope that the author of the program will revise it. To mention just one feature of this program, it will edit in memory one or several files up to the limit of the memory capacity of the computer and then put the overflow on a disk. I once edited simultaneously in memory two versions of a file that were over 130K each! I know of no other editor that would allow that. Together with all keys being redefinable and its other features, this adds up to a super editor.

The article, "The Truth about BASIC" was simple and elegant, but the author missed a trick on the mixing of arithmetic and logical operators. Try the BASIC instruction:

DEF FNMAX(A,B) = A + (B>A AND B−A)

The slowness of interpreted BASIC is such that it hardly matters whether you use "and" or "*," but in a fast language, the difference could be substantial. Multiplication and division are much slower than the logical operators.

Michael Barr
Montreal

There is at least one other editor that handles multiple documents well: P-Edit from SSI. It can edit two documents of arbitrary size, limited only by the capacity of mass storage.

—WF

C—More

Many thanks for the recent series comparing C compilers. Bill Hunt is to be commended for preparing such a detailed check list for comparison among compilers. He presented detailed criteria (with specific subcriteria)—Ease of Use, Language, Standard Library, Expanding Beyond C, Memory Usage, Compatibility with PC-DOS—for making objective comparisons without subjective prejudice. In particular, the multidimensional benchmarks ("Pentathlon" and "100 Yard Dash") are a big improvement over simplistic one-dimensional benchmarks—if I/O transfers are your bottleneck, floating-point times are unimportant, and vice versa.

One technical observation: Hunt implied that compilers that support "register" variables may penalize programs that do not make use of C "registers." This is not universally true; some compilers (Whitesmith's, for example) save and restore C's registers only if the function makes use of them. In any event, C programmers should be encouraged to use "register" when appropriate; it can help and it never hurts.

And one concern about his bottom-line conclusions: Hunt reported a number of deficiencies in Whitesmith's manuals, and he gave this as the reason for an unusually strong negative judgment about its compiler. However, that compiler looked very strong on all the objective criteria. As an educator and writer I favor understandable manuals, but there is a question of balance in application of criteria here. I can report from personal experience (roughly one-third of our training customers have used Whitesmith's compilers) that its products do not deserve the summary dismissal that Hunt gave them. I suggest that an apology, or at least clarification, may be in order.

All in all, however, this article makes a much-appreciated contribution to the usefulness of C language on the IBM PC.

Thomas Plum
President, Plum Hall
Cardiff, NJ

Bill Hunt Replies:

I was grateful for the praise and somewhat concerned with the difference of opinion about the Whitesmith's compiler. To resolve the issue, I talked with Mr. Plum by phone and came up with these results.

Mr. Plum has no experience with the PC-DOS version of the compiler, on which my review is based. Much of the documentation would be more useful in other environments, such as the UNIX or the IDRIS systems that Whitesmith's markets.

I feel that the basic documentation is a necessity for getting up to speed and answering reference questions. Mr. Plum feels that inadequate documentation might be balanced by virtues in other areas. (One such virtue is fast 8087 floating point arithmetic.) Every other product that I tested had acceptable documentation, and I felt that the Whitesmith's compiler did not provide any virtues not available elsewhere.

Mr. Plum says that objective

HOW TO GET AN IBM® PC FOR JUST $1995.

BUY A CHAMELEON.

The Chameleon by Seequa does everything an IBM PC does. For about $2000 less than an IBM.

The Chameleon lets you run popular IBM software like Lotus® 1-2-3™ and Wordstar.® It has a full 83 key keyboard just like an IBM. Disk drives like an IBM. And a bright 80 × 25 character screen just like an IBM.

But it's not just the Chameleon's similarities to the IBM that should interest you. Its advantages should, too. The Chameleon also has an 8 bit microprocessor that lets you run any of the thousands of CP/M-80® programs available. It comes complete with two of the best programs around, Perfect Writer™ and Perfect Calc.™ It's portable. And you can plug it in and start computing the moment you unwrap it.

So if you've been interested in an IBM personal computer, now you know where you can get one for $1995. Wherever they sell Chameleons.

The Chameleon by

SEEQUA
COMPUTER
CORPORATION
8305 Telegraph Road
Odenton, MD 21113

For the location of the Seequa dealer nearest you, call (800) 638-6066 or (301) 672-3600.

CIRCLE NO. 265 ON READER SERVICE CARD

PCnetPLUS™ Diskless Board from ORCHID

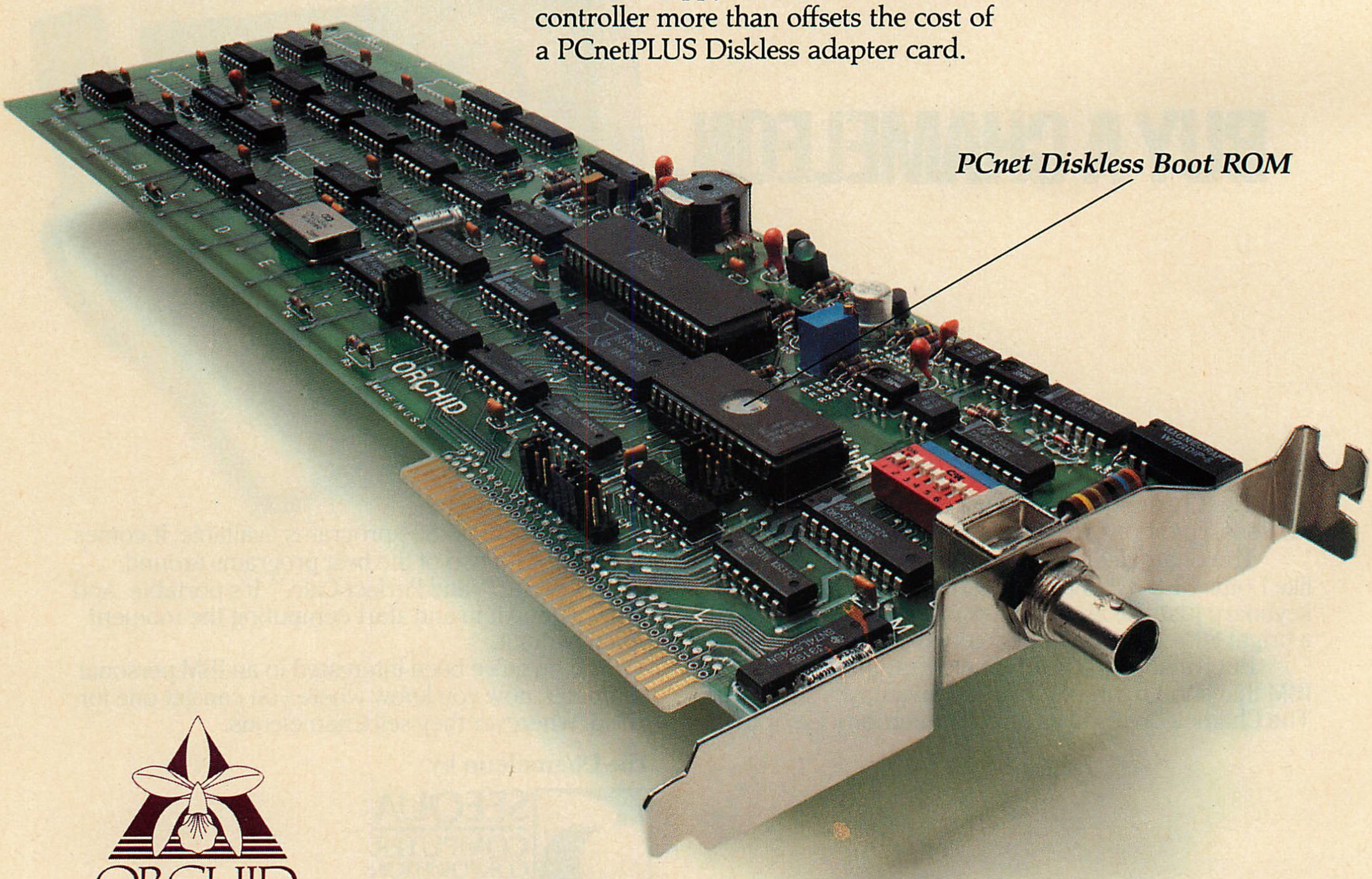
Go with Floppyless PCs

Integrating the leading low cost network, PCnet™ with on-board firmware

Imagine the savings of not having floppy drives, a floppy controller and expensive diskettes at your IBM PCs and compatibles while still maintaining full PC capability. Only Orchid's PCnetPLUS Diskless provides the savings and the many benefits of a floppyless system.

BENEFITS OF A FLOPPYLESS PCnet SYSTEM:

- ▲ **Ease of Operation**—Automatic booting from the PCnet Server, the hard disk unit of the network (i.e. IBM-XT).
- ▲ **Increased Security**—Eliminates the loss of proprietary data and software through the absence of local floppies.
- ▲ **Increased Efficiency**—Elimination of floppy clutter and related diskette maintenance.
- ▲ **Significant Savings**—The elimination of diskettes, floppy disk drives and their controller more than offsets the cost of a PCnetPLUS Diskless adapter card.



AND...PCnetPLUS Diskless includes all the features of PCnet, the leading low cost Local Area Network for IBM-XT/PC and compatibles. Diskless capability is also available on Orchid's PCnetPlus Multifunction Board.

ORCHID

ORCHID TECHNOLOGY, INC.
47790 Westinghouse Drive
Fremont, CA 94539
(415) 490-8586 Telex: 709289

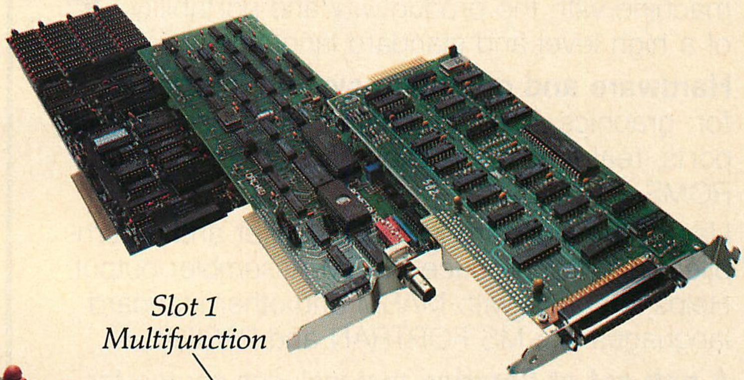
CIRCLE NO. 180 ON READER SERVICE CARD

PCnetPLUS™ Multifunction Board
from
ORCHID

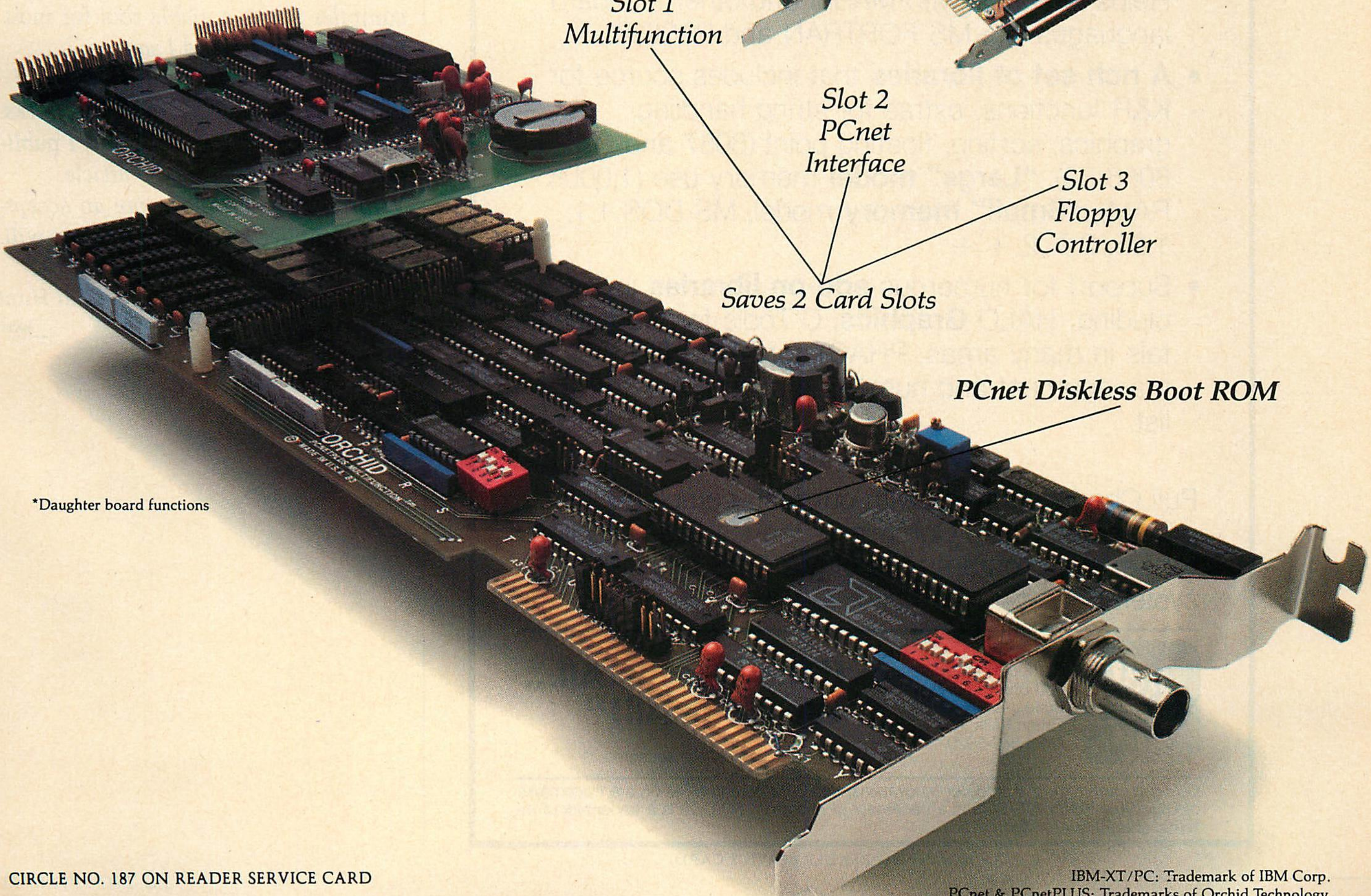
Go with Expandability

Integrating the leading low cost network, PCnet™ with the most popular add-on board

- ▲**PCnet Interface**—The leading Local Area Network for the IBM-XT/PC™ and compatibles.
- ▲**PCnet Software**—Allows PCs to share data, disks, printers, etc.
- ▲**RAM Expansion Sockets**—Up to **256K** of memory.
- ▲**PCnet RAM Drive**—High speed disk emulator software.
- ▲**PCnet SPOOL**—Print Spooler software.
- ▲**EPROM Socket**—For ORCHID supplied **floppyless PCnet** capability.
- ▲**Serial Port***—COM1: or COM2:
- ▲**Parallel Port***—LPT1: or LPT2:
- ▲**Clock/Calendar***—With battery back-up and software.



Take a careful look at a typical PC configuration and see for yourself why you need the PCnet PLUS Multifunction Card.



*Daughter board functions

CIRCLE NO. 187 ON READER SERVICE CARD

measures showed that the compiler compared well. The tables do show that Whitesmith's performance is quite competitive, but the accompanying text discusses several serious problems of the compiler itself: valid C programs were rejected by the compiler for spurious syntax errors and the compiler crashed with perfectly valid option input.

The audience for C compilers on personal computers is quite different from that of several years ago. In the IBM PC environment, many potential buyers are new to C; they have serious applications and are just looking for a good practical programming tool. The Whitesmith's compiler is not adequate for those users. Mr. Plum had in mind an audience more experienced with C and with systems software in general. The compiler is a viable alternative for such buyers.

I write negative comments for two reasons: to save consumers time and money and to goad the vendor into improving serious flaws in his product. I feel that the Whitesmith's compiler is not a viable tool for most potential buyers and I said so. I checked with Whitesmith's to see if improved manuals were in the works. Since they had no such plans, I publicized the problem in the article. Whitesmith's people wrote an acceptable compiler; perhaps now they will provide acceptable documentation.

—Bill Hunt

It's a call from a customer in New York who needs specifics on a business proposal. You talk briefly and then transmit the data to him in seconds from your IBM® PC with PC Exchange.™

Later, an electronic mail message arrives from your Houston office. New developments necessitate an immediate multi-city business trip. Over your IBM PC, you make arrangements with your travel agent. Then you contact an information network for the latest weather in the cities you'll be visiting.

All through your IBM PC. All made possible by PC Exchange—a versatile, dynamic and interactive communications package on a single pc board. Complete with software and tutorials.

With PC Exchange your IBM PC becomes far more than a computer.

PC Exchange: from the people who brought you PC Express.

PC Express was welcomed by PC users everywhere. It packed a number of valuable communications capabilities onto a single pc board occupying just one expansion slot.

Now we've made a good thing even better. We took all the PC Express capabilities, added some exciting new ones, and gave the entire package a new name: PC Exchange.

Through it, you can exchange voice and data communications—directly or remotely—with other IBM PCs, PC-compatible computers and mainframes.

For example, with PC Exchange, your PC can perform DEC™ VT100/52 emulation and allow you to access all major information networks such as THE SOURCE.℠ Membership in THE SOURCE is included with your purchase of PC Exchange.*

PC Exchange also puts at your command a powerful, fully-integrated telephone management system which includes a full-screen text editor, an "Electronic Phonebook," electronic mail capability, an onboard modem, auto-dialing, auto-answering and more.

Available now: SNA 3270 and 3770 RJE.

If your company has an IBM mainframe, chances are it's being overworked and yet, underutilized. People just can't get the data they need when they need it.

The answer: PC Exchange with SNA. With it, your PC can communicate with your IBM mainframe, and emulate a 3274 cluster controller, 3770 Remote Job Entry station, and 3278/9 terminals with multi-session capabilities.

Multi-sessioning gives you the ability to be connected concurrently to four different mainframe applications and toggle back and forth, without logging on and off.

For MIS managers, connecting PCs to mainframes is an effective, cost-efficient way to reduce host processing loads. So, PCs can become an integral part of the corporate information environment.

For a free demonstration, contact your nearest PC Exchange dealer. For the address, call 800-523-8396. See what a dramatic change PC Exchange can make in the speed and efficiency with which you gather and use information.

Intelligent Technologies International Corp., 151 University Avenue, Palo Alto, CA 94301, (415) 328-2411. 800-523-8396. Telex: 756175.

PC Exchange from Intelligent Technologies™

a logica company

CIRCLE NO. 209 ON READER SERVICE CARD

all for you IBM PC."

To every software developer who'd written off portability as an impossible dream, Digital Research humbly announces a few monumental breakthroughs.

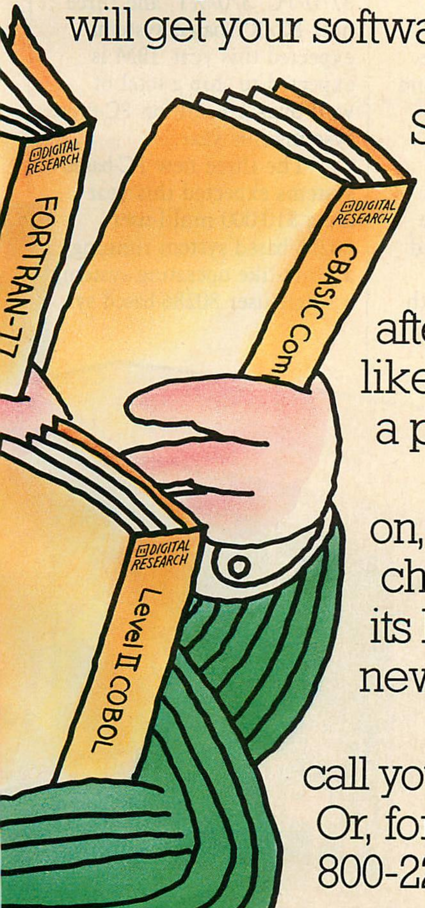
We not only offer languages that are portable from 8 to 16 to the 32-bit chips of the future, they're portable across all popular operating systems, too. What's more, we supply the broadest range of quality languages and development tools available today. And will tomorrow.

So rest assured. Whether you design applications at a major corporation, plan to become a major corporation or just qualify as a hobbyist, you only have to write it once.

Simply pick the Digital Research language that's right for you. From Personal BASIC™ to Digital Research FORTRAN-77.™ The newest member of our remarkable family.

To complement languages, we offer a complete workshop of development tools. Our Display Manager™ and Access Manager™ simplify the design of screen displays and data bases. So you spend less time and effort.

If you write in COBOL, our Animator™ source level debugger will get your software running in record time.



And for programmers skilled with IBM mainframe SPF, we offer micro/SPF.™ An editor that helps turn your invaluable experience into valuable new software applications.

At Digital Research, we work as hard for you after the sale as we do to get the sale. With backup like quality documentation, software updates and a phone line to our technical support team.

With so much productivity and service to draw on, it's small wonder IBM chose our languages for its IBM® PC, XT and the new IBM 3270/PC.

For more information, call your IBM representative.

DIGITAL RESEARCH®
We make computers work.℠

Or, for the Digital Research retailer nearest you, call 800-227-1617, ext. 400. In California, 800-772-3545, ext. 400.

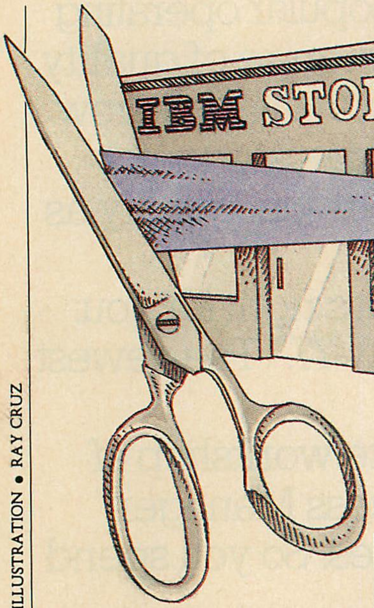


ILLUSTRATION • RAY CRUZ

RANDOM RUMORS AND GOSSIP

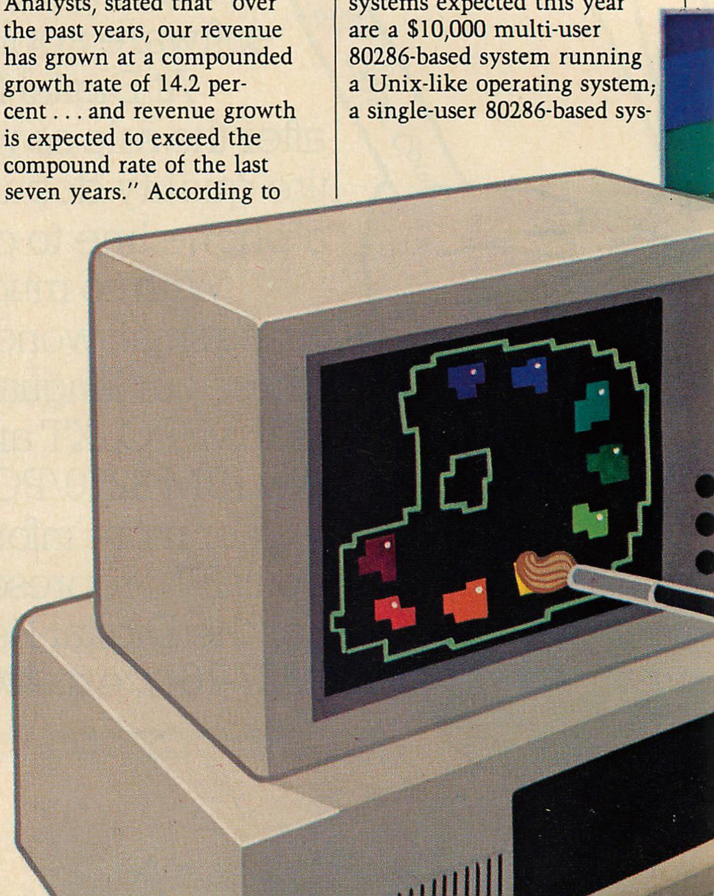
IBM has stepped up the opening of **Product Center stores** and is expected to have close to 600 by the middle of next year. The company has also begun opening retail stores in Europe; four are already open, and there should be about 60 by the middle of next year. **Morrow Designs** is rumored to be readying an **IBM-PC compatible portable** with a display capable of showing 16 lines of 80 characters each. The unit will weigh only 10 pounds. **IBM** is rumored getting ready to replace the 5520 and Displaywriter small office systems with a **new version of the PC** that runs the same software as those systems run. Industry experts are predicting that **IBM** will soon reveal a **Unix-like operating system** that will run as a "task" under their CMS (Conversational Monitor System) on 370/XT- and 4300-class machines. **IBM** is also believed readying two new low-end **4300-type machines** using Motorola 68000 microprocessors. These machines, expected to sell for about $45,000, should perform at about 0.3 MIPS and serve as hosts in small office systems. They should run shrunken versions of VM and CMS operating systems. There are predictions that **IBM** will ship some 35,000 **370/XT systems** this year. The **IBM 9000 series machine** (68000-based), from the IBM Instrumentation Division, now has a Unix operating system (Xenix) and is expected to become a commercial product available via IBM's National Accounts Division. **Lotus Development** has delayed introduction of its **new version of 1-2-3**, originally promised for January. Rumors are that it will not be available until June, at the earliest. The new version is expected to have word processing and telecommunications facilities. Lotus is also rumored working on versions of 1-2-3 to run on Motorola 68000-based machines and on the new IBM 370/XT. **IBM** is supposedly pushing back introduction of its **Local Area Networking system** because of problems with deliveries of chips from Texas Instruments. IBM demonstrated a prototype version of the baseband system last October and was expected to begin beta-testing systems with the TI chips this spring. The introduction of this system may now be pushed back to the end of the year. **Tecmar** is rumored readying a 68000-based CPU card for IBM PC-compatible machines and will include with it a version of the UNIX-like Coherent operating system.

IBM EXPECTS MORE THAN 14 PERCENT REVENUE GROWTH IN 1984

Allen J. Krowe, senior vice-president of IBM finance and planning, speaking at a recent meeting of the New York Society of Security Analysts, stated that "over the past years, our revenue has grown at a compounded growth rate of 14.2 percent . . . and revenue growth is expected to exceed the compound rate of the last seven years." According to Krowe, IBM anticipates the continuation of a "remarkable growth" for its personal computer business. In addition, "in 1984, worldwide shipment of personal computers should be more than three times the 1983 shipment level. . . . The personal computer gross profit margin is obviously lower than some larger products, but our efforts in lowering distribution costs are paying off. The operating margin has been growing and is excellent for a product with such a high asset turnover."

Analysts estimate that IBM shipped close to 850,000 PC/XTs last year. This year, they estimate, IBM will double that number and also produce about 1 million PCjrs. Added to this are the 3270/PC, 370/XT, and three new PC/XT-based systems expected this year. IBM is expected to ship a total of well over 2.5 million PC-type systems this year.

The three new PC-based systems expected this year are a $10,000 multi-user 80286-based system running a Unix-like operating system; a single-user 80286-based sys-



THE TECH JOURNAL NEWSLINE

News, views, and gossip on the IBM and IBM-like marketplace

tem to compete with the Apple Lisa; and a portable version of the PC that may include some bubble memory.

EMI-RESISTANT VERSION OF PC UNVEILED

IBM has begun, very quietly, to market a special version of the PC/XT that is EMI-resistant. The system has a special shielded enclosure and shielded cables that prevent it from radiating any radio frequency signals that could be picked up by surveillance systems. The system would also be resistant to induced RFI (Radio Frequency Interference). Reportedly, the system is made to conform to a secret specification standard from the National Security Agency called "Tempest" (NACSIM spec 5100A). It is expected that these systems will be purchased by the Central Intelligence Agency, the National Security Agency, and the military. The system is available from IBM's Federal Systems Division in Gaithersburg, MD, for about $7,000.

NEW GRAPHICS SYSTEM FROM IBM

IBM has unveiled a new graphics system that can produce images with up to 256 hues of color or 256 gray shades with a density of one million pixels. Called the IBM 5080, it has a 19-inch screen, tablet, and either a mouse or a stylus for drawing. The system is designed for scientific and engineering CAD/CAM use. Pricing starts at $20,000.

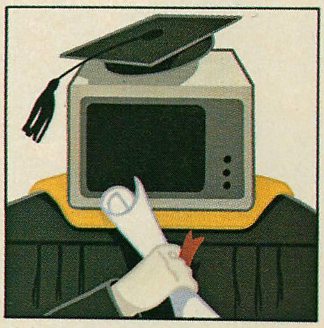
LOTUS & VISICORP ANNOUNCE 3270-COMPATIBLE SOFTWARE

Both Lotus Development Corporation and VisiCorp have announced versions of their spreadsheet programs that are compatible with the newly unveiled 3270 version of the IBM PC. The 3270/PC was announced in November, and demo units have been in IBM sales offices since December; customer shipments are expected to start shortly.

The 3270/PC emulates the classic IBM 3270 intelligent terminal used with IBM mainframe hosts. It has a high-resolution color display with windowing capability. A user can run several tasks on the mainframe and a standard IBM-PC task on the 3270 concurrently, with each task displayed in a window.

Lotus's new version of 1-2-3 will run in a 3270/PC window and utilize the 3270/PC keyboard, which is different from that of the PC/XT. VisiCorp has released a special version of VisiSeries and VisiOn software that they say "takes advantage of the [3270/PC] system's capabilities." These versions are expected to be sold only through national accounts sales forces, not through retail channels.

Both companies are also expected to introduce versions of their spreadsheets to run on the new IBM 370/XT machines under the CMS operating system.



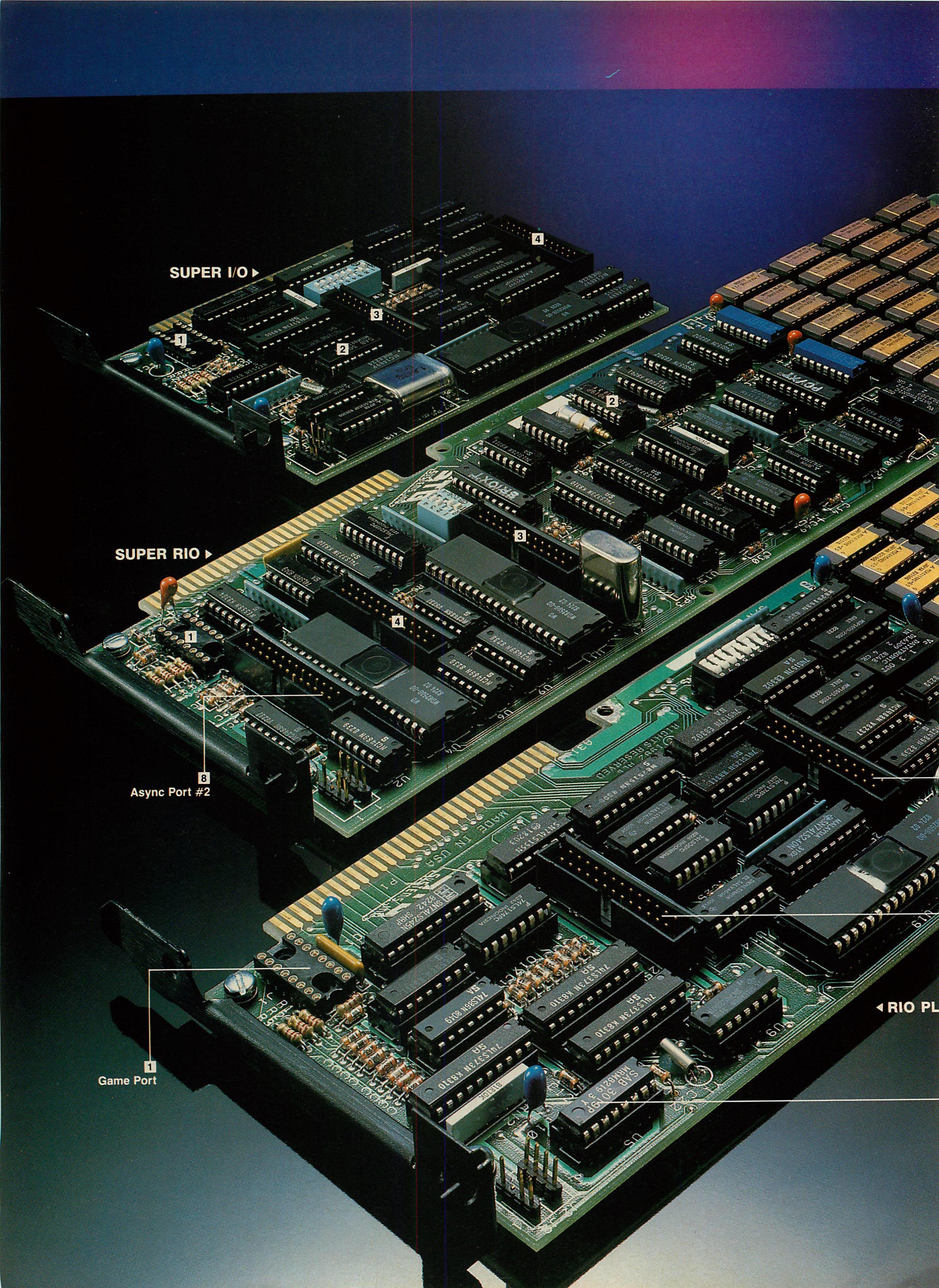
PCjr GOES TO SCHOOL

Virginia Polytechnic Institute's College of Engineering has signed a contract with IBM for 1,600 PCjr's, expandable to 4,000 units, for VPI's 6,000 students and 250 faculty members. Beginning in September, all freshmen will be required to buy a PCjr at a discount. The school also ordered 300 PCs.

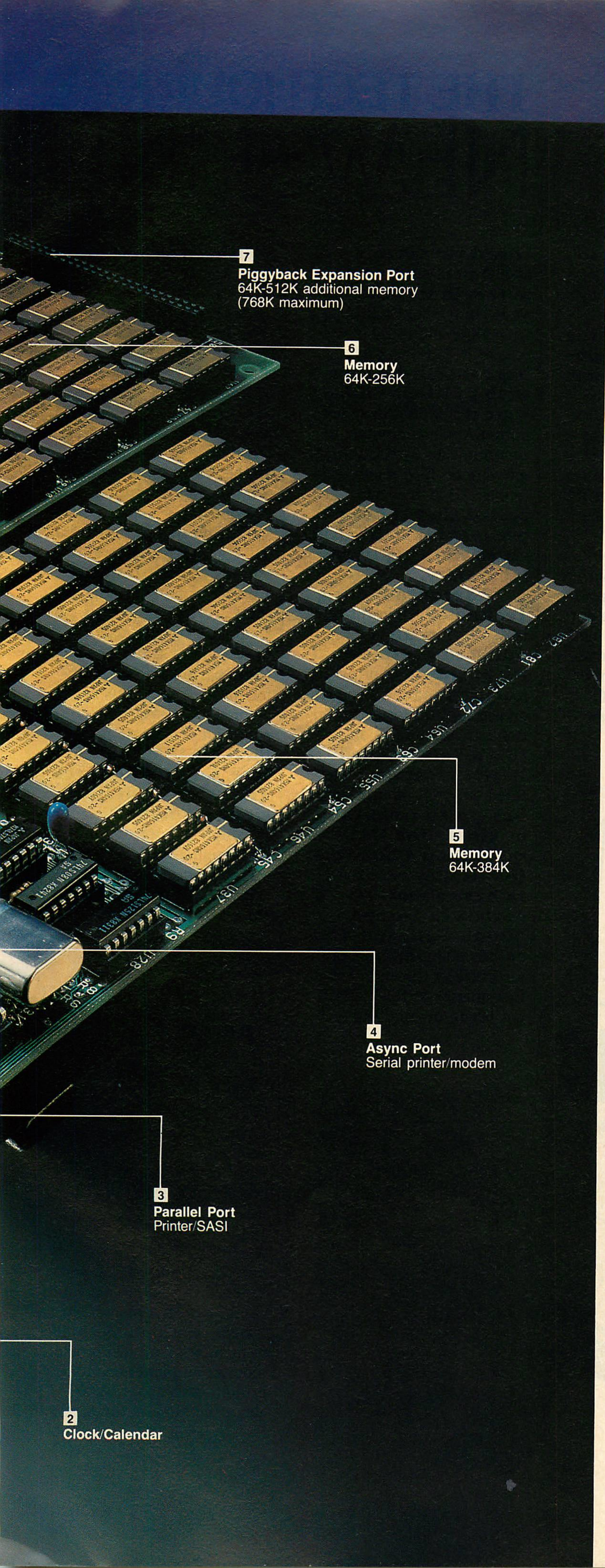
RANDOM NEWS BITS

Lotus Development Corporation has purchased a license for the **Microcom Networking Protocol** (MNP) from Microcom Inc. in Norwood, MA. MNP enables personal computers to communicate with other computers supporting MNP. Among those companies that have also purchased MNP licenses are VisiCorp, Apple Computer, and GTE Telenet. **Microsoft** is now reported to have **licensed MS-DOS** to almost 100 OEMs. **Digital Research** reportedly has **licensed CP/M** to nearly 300 OEMs (85 percent of these licenses were for the

ILLUSTRATIONS • CAMERON WASSON



SUPER I/O ▸
SUPER RIO ▸
8
Async Port #2
1
Game Port
◂ RIO PL



What you get out of your IBM PC/XT depends on what you put into it.

The versatile multi-function boards from STB offer more functions and more memory.

If your personal computing applications have outgrown your personal computer's hardware, there's a smart way of expanding your system while still leaving plenty of room to grow.

The secret is the line of multi-function boards from STB.

Our SUPER RIO™ combines two asynchronous RS-232-C serial communication ports for interfacing modems, serial printers, plotters, mice, and an almost limitless list of other peripherals, a parallel I/O port for interfacing printers and hard disk controllers, a battery operated clock/calendar that remembers the date and time even when your system is turned off, a game paddle port that can uses either APPLE™ or IBM™ compatible joysticks, and up to 256K of additional memory.

The STB Piggyback™ board adds up to 512K more memory onto the SUPER RIO for a total of 768K. And the pair uses only one expansion slot.

Included free with the SUPER RIO board is the PC Accelerator™, which combines printer buffering and high speed disk emulation into one easy to use program.

Our RIO PLUS™ board includes all of the I/O functions of the SUPER RIO (less one serial port) with on board memory expansion capability of up to 384K. When used with a 256K computer, this board gives your system a full 640K of memory. Of course, PC Accelerator is also included with every RIO PLUS.

If you already have enough memory, the STB SUPER I/O™ board adds a parallel I/O port, serial port, game port, clock/calendar and PC Accelerator to your system. And, its small enough to fit in a short slot on the XT.

As new functions and applications arise, you can count on STB Systems to provide convenient, affordable ways to adapt these innovations to your system.

Compare our quality products at your local dealer, or for more information please write or call us.

Expanding Microcomputing

STB™

STB Systems, Incorporated

601 North Glenville, Suite 125, Richardson, Texas 75081/(214) 234-8750

All STB multi-function boards include a serial and parallel cable plus an external mounting bracket.
Apple is a registered trademark of Apple Computers, Inc.
IBM is a registered trademark of International Business Machines Corporation.
PC Accelerator is a registered trademark of ResiCorp.

CIRCLE NO. 199 ON READER SERVICE CARD

eight-bit version; 15 percent for the 16-bit version). **IBM** has begun to manufacture **floppy and hard disk drives** in-house. The expectation is that this is to supplement purchases from vendors such as Seagate Technology, MiniScribe, Tandon, Control Data, Qume, and International Memories. **Digital Research** has released a Fortran-77 compiler that the company claims is faster than currently available Fortran compilers for micros.

IBM has begun letting software-only stores sell IBM-brand software. The first such chain (32 stores) is Software Centres International of Los Angeles. **There are now over 800 PC dealers in 26 countries** outside North America. Judging by companies exhibiting at trade shows, **the training of end-users is becoming a big business,** with close to a dozen companies already in the field. **Trilian Computers,** Box 481, Los Gatos, California 95031, **has introduced a visual shell for the PC called "VisuAll."** The shell allows the user to create "pop-up" VisiOn-like menus that will work within existing software packages and uses a mouse or the cursor control keys. It does not need a hard disk and works with only 128K of memory. Watch out VisiOn, DesQ, and Lisa! **DEC has disclosed that it sold 50,000 personal computers in its first year,** only about a fifth of what IBM sold in its first year of PC sales. **Apple has cancelled a pact with Cullinet software** that would have permitted Apple computers to communicate with Cullinet software on IBM mainframes. **IBM has reported that it has experimental 512K RAM chips working in the laboratory.**

Advanced Matrix Technology, Newbury Park CA, **has announced a letter-quality printer** that prints in four colors and can also print on acetate film for overhead transparency projection. **IBM's Value-Added Dealer program,** instituted only ten months ago, **has grown from the original nine VADs to more than a hundred. IBM is now reported to have almost 800 PC dealers in Europe, Middle East, and Africa;** the company is shooting for 1,000 by the end of the first quarter of 1984. **IBM** has begun distributing the **VisiOn** operating environment and applications programs for the PC/XT. The question now is what will IBM do when microsoft releases its new window extension to MS-DOS as early as next month? **Intelligent Systems Corp** (ISC) of Norcross, GA, has acquired **Princeton Graphic Systems** (PGS), a manufacturer of high-resolution IBM-PC-compatible color monitors for $15 million in stock. PGS is ISC's sixth subsidiary. Another is **Quadrum Corp.,** a maker of plug-in cards for the PC/XT. **Imagic,** of Los Gatos, CA, is reported to have shipped some of its cartridge-based game software for the PCjr to dealers even before dealers had units to run it on. There are reports that when **IBM's San Jose, CA, and Research Triangle, NC, divisions** had trouble obtaining PC shipments for test and measurement use, they went outside and purchased PC-type main computer boards from Faraday Electronics.

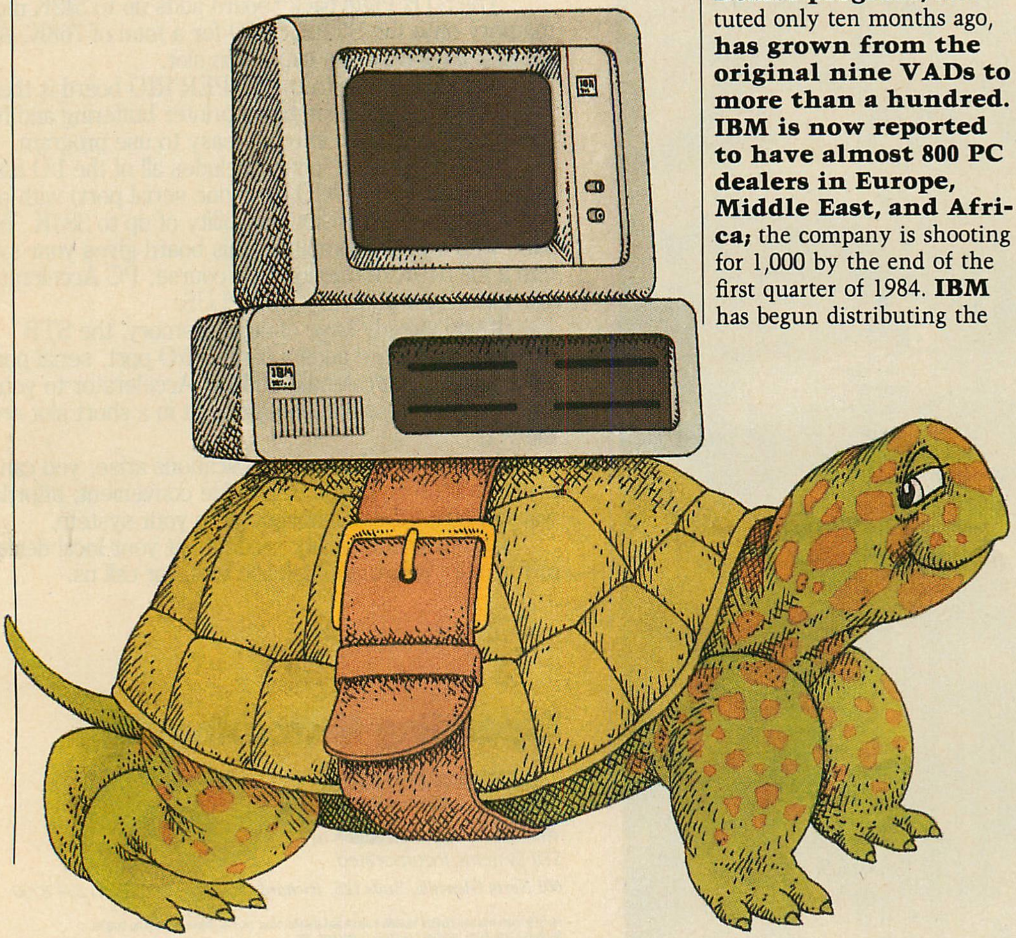
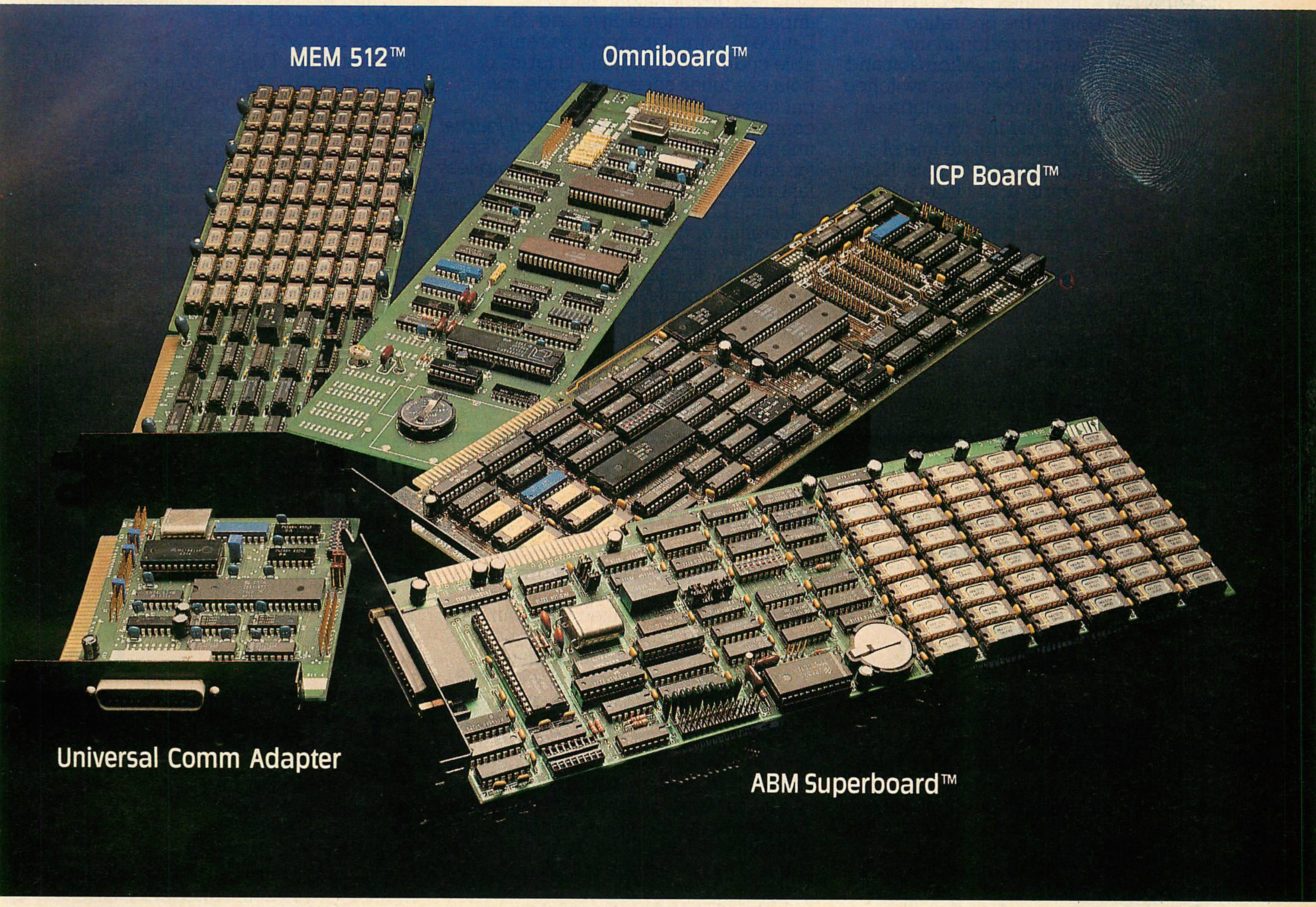


ILLUSTRATION • RAY CRUZ

VISION FINALLY ARRIVES

Thirteen months after it was first demonstrated VisiCorp's VisiOn system for the IBM PC/XT is now being shipped. The question now is whether the long delay will hurt its chances for success. Initial reports indicate a lukewarm response. Dealers and users are comparing it to Apple's Lisa, which they find significantly faster and easier to use. Some users also point out that a basic VisiOn system, which requires 512K of memory, hard disk, color monitor and letter-quality printer, costs about $8,000, about the same as the Lisa.

Build A Better PC



With ABM's complete line of PC enhancement products, you can **build a better PC.**

ABM Superboard™ provides zero to 384K bytes of RAM memory, serial port, parallel port, IBM or Apple compatible game port, and clock/calendar with battery backup and external alarm feature. ABM Superset utility software allows you to configure part of memory for print spooling, so you can continue computing while your printer is busy, and for RAM disk, which emulates a disk drive and dramatically increases your productivity.

Omniboard™ provides up to two serial ports, parallel port, clock/calendar with battery backup, Apple compatible game port, and Superset utility software.

MEM 512™ memory board provides 64K to 512K bytes of field-upgradeable RAM memory on a single board. ABM Superset utility software is included.

Col-Mon™ ABM's color to monochrome adapter permits display of full text and graphics from the IBM PC color graphics board on the monochrome monitor.

ICP Board™ Intelligent Communications Processor is the leading edge solution for remote PC to IBM mainframe communications. Utilizing a Z80-B microprocessor with 64K bytes of RAM memory to run 3274/3276/3278 emulation software, it eliminates the need for an IBM cluster controller, and maintains the host link during local PC processing. ICP cluster capability allows attachment of up to 12 more PC's, offering unparalleled cost advantages.

Universal Comm Adapters ABM's dual channel comm adapters can support ASYNCH, BSC, SDLC, HDLC, X.25, and other communications protocols. 3270 SNA/SDLC support for PS-SC2 is available from ABM.

Warranty All of ABM's products are rigidly tested and burned in. This allows us to offer a complete one-year warranty at no additional charge. Check our features, our quality—then ask your PC dealer for Personality Series products from ABM Computer Systems.

Dealer and OEM inquiries welcome.

ABM

ABM Computer Systems
3 Whatney, Irvine, California 92714
(714) 859-6531 TWX 292427 ABM UR

Personality For Your PC

CIRCLE NO. 273 ON READER SERVICE CARD

PC BRAND Can Make You

C has rapidly become the language of choice for serious software developers. The language of maximum portability across the spectrum of today's personal computers. The language which will most rapidly adapt to tomorrow's machines. The companion language to Unix™, the operating system destined for predominance. Major companies such as Microsoft and Micropro know this. They have switched to C for their applications development. How ready are you for the future?

PC Brand has the complete line of today's finest professional C tools. Not just the finest compiler, but libraries for graphics and screen design, overlay linkers, and program editors acclaimed as the best of their kind. It is a complete workshop of craftsman's tools which will turn you into a C professional in a hurry; capital equipment which will pay back your investment many times over.

Take the time to read this advertisement. You are sure to see the competitive edge these products will deliver.

LATTICE C

The Preeminent 16-bit C Compiler

You can find a more economic way to learn C, but if your interest is program development, Lattice C™ is the unparalleled choice. *Byte* said "the Lattice C compiler produces remarkable code . . .outstanding in terms of both execution speed and code compactness". After reviewing nine compilers for the PC, the *PC Tech Journal* unequivocally declared Lattice C "best for software development. . .it compiles fast and produces fast programs".

Lattice C is a full implementation of Kernighan and Ritchie, not a subset, and even offers extra features such as nested comments, 39-character variable names and extra compile time checks for some of C's subtler errors. The compiler comes with a full library of I/O routines which implement under MS™-DOS most of the Unix-compatible standards described by Kernighan and Ritchie.

Lattice C runs on virtually any computer using an 8086 or 8088 microprocessor, and we carry two versions for either PC-DOS™ or CP/M-86™. Create your source files with any word processor or text editor like Edlin or our Pmate™ and Lattice C will compile them into Intel 8086 object module format ready for linking with other modules by linkers such as DOS' Link or our Plink86™.

Lattice C offers a choice of four memory models which allow the program designer to choose the right combination of efficiency and size for an application: a range between 64K and a full megabyte for program and data area size.

The documentation, which *Byte* says "sets such a high standard of excellence that others don't even come close", features sample source programs and covers the interface to assembly language and machine dependencies.

C's structured approach encourages development of tight, fail-safe functions which can be counted on to return reliable results every time. Local variables unknown outside of functions to safeguard against collision. Extremely powerful nested expressions which produce elegant, concise code.

Requires 128K RAM.

For PC-DOS: Product Code: S0100
For CP/M-86: Product Code: S0120
Suggested Price: **$500.00**
Our Price: **$400.00**

HALO

A Spectacular Graphics Extension to Lattice C

Halo™ will astound you. It provides a complete library of graphic functions which can be linked with your Lattice programs to create full-color charts, graphs, simulations, even animation.

Over 100 commands are at your disposal, including plot, line, arc, box, circle, plus single commands to produce bar graphs and pie charts. Pattern-fill and dithering commands give your graphics impressive texture and color mixes, rubber-banding draws shapes for interactive users, area moves produce animation, fill and flood commands paint areas. The newest version allows you to define your own world coordinate scheme, divide the screen into "viewports" (windows), and scale graphic figures automatically.

It's a long list of capabilities which make for an extraordinarily powerful product. In fact, Halo is so good that manufacturers of graphics boards and systems are adopting it as a standard graphics language. So it can bridge your application to other systems. CAD-CAM developers, especially, have embraced its device-independent approach for maximal portability.

Halo is a dazzling demonstration of why C has become the language of choice among programming professionals: its function library architecture means you can tremendously enhance your firepower by acquiring libraries of software like Halo with dramatic economy of time and money.

Requires IBM monochrome or color graphics card or equivalents.

Product Code: S0300
Suggested Price: **$200.00**
Our Price: **$160.00**

PMATE

The Programmer's Word Processor

Pmate was designed for programmers. We'll wager that you cannot find a programmer who has discovered Pmate and moved on to something else.

Pmate is a full screen editor with ten auxiliary buffers for squirreling away pieces of text until needed. It uses single key commands to move the cursor, or text, or insert or delete, or rescue several thousand characters of deleted text.

It has a format mode for tab setting or wraparound and shaping when it's time to write documentation. Pmate lets you assign chains of commands or strings of text to single keys: one keystroke could set up the entire shell of a new C function, for example.

Pmate has variables, if-then statements, loops. It calculates, and converts decimal to hex to binary and back. You can write compact programs (called "macros") to delete comments, for example, or check syntax, or process long sequences of commands. Macros can alphabetize lists, do row and column math, perform a series of operations on multiple files, even summon other macros.

Put another way, Pmate is a text editor with its own built-in interpretive language. A language you can use to completely customize this text editor to your fancy. Possibly the most artful, ingenious program you have ever seen.

Product Code: S0600
Suggested Retail: **$225.00**
Our Price: **$180.00**

PLINK86

Overlay Linkage to Expand

Software is becoming ever more sophisticated, which means more complex programs requiring large chunks of memory. But if you use extra memory, if you count on users to have expanded RAM, you will forego sales to those who do not.

Plink86 is the answer. It takes on the job of shoe-horning large programs into small memory. First, Plink86 acts as an alternative to DOS' Link. For a language like C which encourages design of separately compiled object modules in the Microsoft relocatable format, Plink86 pulls modules together into single compiled programs. But Plink86's overlay power is what has gained it a reputation as a miracle worker. It binds into the compiled program its overlay manager which knows how to swap modules of your large linked program between disk and memory, so that each can temporarily occupy the same memory space.

Unlike other linkers, the overlay manager acts on its own, needing no calls from the source program. Instead, Plink86's straightforward overlay description language allows you to describe your overlay structure in one place in your program — a structure

ELIEZER NADDOR

AFTER BUBBLESORT:

sorting methods and timing on the IBM PC

8 Knuth sorting algorithms translated into BASIC

Until 1973, I knew very little about sorting, mainly because I had little need to sort large quantities of numbers or records. If one of my programs needed some sorting, I would use the easy-to-code (but very slow) Bubblesort, as many people still do.

In 1973, Addison-Wesley published Volume 3 of Donald E. Knuth's *The Art of Computer Programming*, which is devoted to "Sorting and Searching." This book is not too easy to read, but every moment spent on it has great payoffs. Knuth is one of the giants of computer science. He is a great mathematician, a master programmer, an excellent teacher, and a very thorough historian of mathematics and computing.

Knuth presents in his book some 25 sorting algorithms. For each algorithm he gives "(a) an English-language description of the algorithm, (b) a flow chart, (c) a MIX program, and (d) an example of the sorting method applied to a given set of numbers." (p. 75). MIX is his unique assembly language, which is defined in Volume 1 of his book.

In order to make Knuth's algorithms accessible to non-computer science students, I translated six algorithms into BASIC, and made them part of a single program called SORTER. The program was initially implemented on a PDP-11/45 and was then transferred to a DEC-10. In 1978, one

ENTER . . . THE FUTURE!

Introductory offer

$49.95

THIS IS THE PASCAL COMPILER EVERYBODY'S BEEN WAITING FOR... EVERYBODY EXCEPT THE COMPETITION!

	Turbo Pascal	IBM Pascal	Pascal MT+
PRICE	49.95	300.00	595.00
Compile & Link speed	**1 second!!!**	97 seconds	90 seconds
Execution speed	**2.2 seconds**	9 seconds	3 seconds
Disk Space 16 bit 8 bit	**33K w editor!** **28K w editor!**	300K + editor Not Available	225K + editor 168K + editor
8 and 16 bit	**YES**	NO	YES
built-in editor	**YES**	NO	NO
Generate object code	**YES**	YES	YES
One pass native code compiler	**YES**	NO	NO
Locates RunTime errors directly in source code	**YES**	NO	NO

Benchmark data based on EightQueens in "Algorithms + Data Structures = Programs" by N. Wirth, run on an IBM PC.
Turbo Pascal is a trademark of Borland International. MT+ is a trademark of MT MicroSystems. IBM is a trademark of International Business Machines.

LEARN TO WRITE A SPREADSHEET

Our Introductory offer includes MICROCALC, a spreadsheet written in **Turbo Pascal.** It will be on your disk, and ready to run. And we've included the source code to show you exactly how a spreadsheet is written!

[redacted]

Turbo Pascal includes a 250 page bound manual with extensive explanations and many illustrative examples.

Extended Pascal for your IBM PC, APPLE CP/M, MS DOS, CP/M 86, CCP/M 86 or CP/M 80 computer features:

- Full screen interactive editor providing a complete menu driven program development environment.
- 11 significant digits in floating point arithmetic.
- Built-in transcendental functions.
- Dynamic strings with full set of string handling features.
- Program chaining with common variables.
- Random access data files.
- Full support of operating system facilities.
- And much more.

ORDER YOUR COPY OF **TURBO PASCAL** TODAY TO TAKE ADVANTAGE OF OUR INTRODUCTORY SPECIAL.
For Visa and MasterCard orders call toll free

1-800-227-2400 X 968

IN CA: 1-800-772-2666 X 968

(lines open 24 hrs. a day, 7 days a week)

Dealer & Distributor Inquiries welcome.

Turbo Pascal $49.95 + $5.00 shipping per copy.

Check______Money Order______
VISA______Master Card______

Card #:________________
Exp date:__________Shipped UPS

BORLAND INTERNATIONAL

Borland International
4807 Scotts Valley Drive
Scotts Valley, California 95066

My system is: 8 bit______16 bit______
Operating system: CP/M 80__________
CP/M 86____MS DOS____PC DOS____
Computer:______Disk Format:________
Please be sure model number and format are correct.

NAME: ____________________
ADDRESS: ____________________
CITY/STATE/ZIP: ____________________
TELEPHONE: ____________________

California residents add 6½% sales tax. Outside North America add $15.00. Checks must be on a U.S. bank, and in U.S. dollars. Sorry, no C.O.D.

See Software.

Dick is a programmer. Dick is bored. Harried. Overworked. Dick struggles with tedious trace chores and debugging routines. Non-existent documentation. Hidden bugs. Dick is four months behind schedule as a result. And customers are angry when bugs slip through. They yell and make Dick upset. They make Dick's boss upset. Nobody is very happy.

Dick dreams of a different sort of life. Where he's a programmer hero. Entertained by his work. Admired for his skill. Rewarded for his performance. Now his dreams can come true.

See Software Run.

Jane is a happy programmer. She uses ANIMATOR.™ It's a unique Visual Programming™ aid for Micro Focus COBOL. It runs on Jane's friendly microcomputer. It makes child's play of test and debugging tasks.

With ANIMATOR Jane sees a picture of the program explaining itself. In live action. In real time. In COBOL source code. As ANIMATOR displays the program listing, the cursor tracks the exact execution path. Including subroutine branches.

of my colleagues, Rainer von Saleski, added two other sorting algorithms to the program. Recently, I recoded the entire program for a VAX/VMS and for an IBM PC. The implementation and performance of the sorting algorithms on the PC are the main subjects of this article.

Eight Sorting Algorithms. Following is a very brief description of the eight algorithms.

SELECTION (Knuth, pp. 139-142): "First the smallest . . . is selected; then the next smallest is selected, and so on" (p. 75).

BUBBLESORT (Knuth, pp. 105-110): "If two items are found out of order, they are interchanged. This process is repeated until no more exchanges are necessary." (p. 75).

INSERTION (Knuth, pp. 80-82): "The items are considered one at a time, and each new item is inserted into the appropriate position relative to the previously-sorted items." (p. 75).

SHELLSORT (Knuth, pp. 84-95): The algorithm is based on an article by D. L. Shell, *CACM* (Vol. 2, July, 1959 pp. 30-32), and is a simple but powerful extension of INSERTION.

QUICKSORT (Knuth, pp. 114-123): The algorithm is based on an article by C.A.R. Hoare, *Comp. J.* (Vol. 5, 1962, pp. 10-15). The items are divided into two groups, in which every item in the first group is smaller than every item in the second group. A similar division is then made in each of the two groups and so on until each group has only one item.

SINGLETON (Knuth, p. 123): This is an extension of the QUICKSORT algorithm and is based on an article by R.C. Singleton in *CACM* (Vol. 12, 1969, pp. 185-187). Von Saleski initially coded Singleton's approach and added several additional refinements.

TREESORT (Knuth, p. 168): This is based on an article by L. J. Woodrum, "Internal Sorting with Minimum Comparing," *IBM System Journal* (Vol. 8, No. 3, 1969, pp. 189-203), and on its BASIC coding by Richard Hart, "A New Fast Sorting Algorithm," *Creative Computing* (Jan-Feb, 1978, pp. 96-101). ". . . The algorithm was written to use as few comparisons as possible, to have as few steps between each comparison as possible, to take advantage of natural sequencing, . . . " (Hart, p. 96).

HEAPSORT (Knuth, pp. 145-149): This is based on an article by J. W. J. Williams, *CACM* (Vol. 7, 1964, pp. 347-348). "First we rearrange the file until it forms a 'heap,' then we repeatedly remove the top of the heap and transfer it to its proper final position." (p. 146).

KNUTH'S ANALYSIS OF SORTING ALGORITHMS

In his book, Knuth makes a very detailed analysis of most sorting algorithms.

> *Suppose it takes 25 seconds to sort 64 names. It would be logical to expect BUBBLESORT to take 25*16=400 seconds to sort 256 names, and 25*256=6400 seconds to sort 1024 names.*

For SELECTION, BUBBLESORT, and INSERTION he shows that, on the average, the expected time to sort N items is CN^2 (pp. 141, 109, and 82, respectively). The constant C depends on the algorithm, and, as we shall see, on several other factors related to the PC. The effect of the constant C can be eliminated if one compares the times for two sets of N1 and N2 items, using the same algorithm under identical circumstances. If N2 is larger than N1, then the expected increase in time is $(N2/N1)^2$. In order to appreciate the significance of this quantity, consider figure 1 and, say, the BUBBLESORT algorithm. Suppose it takes 25 seconds to sort 64 names. One would then expect that BUBBLESORT would take 25*16 = 400 seconds to sort 256 names, and 25*256 = 6400 seconds to sort 1024 names.

Knuth also shows that the expected time to sort N items with QUICKSORT, SINGLETON, and HEAPSORT is approximately C*N*log(N) (pp. 121 and 149), where C is a constant depending on the algorithm and other factors. The increase in time from N1 to N2 items for the same algorithm and under identical circumstances is therefore expected to be (N2/N1)*log(N2)/log(N1). The values of this quantity for various N1's and N2's are given in figure 1A. Consider, for example, the QUICKSORT algorithm, and assume that it takes 7 seconds to sort 64 names. One can then expect that QUICKSORT would take 7*5.33 = 37 seconds to sort 256 names and 7*26.67 = 187 seconds to sort 1024 names. As we shall see below, all these expected numbers of seconds for BUBBLESORT and QUICKSORT are quite close to the actual times. BUBBLESORT is very, very slow, and QUICKSORT is very, very fast.

The expected time for SHELLSORT appears to be difficult to analyze (p. 91). Knuth shows that this time is less than $CN^{1.5}$ (pp. 91-92) and mentions the times $CN^{1.28}$ to $CN^{1.25}$ in connection with certain assumptions and experiments (p. 92). For the range of 32 to 1024 items, these quantities are somewhat close to C*N*log(N). We have therefore added SHELLSORT to the list of algorithms to which figure 2 applies.

Knuth does not discuss the TREESORT algorithm in his book—he only cites Woodrum's paper. From the results cited below in "Timing Results on the PC," it appears that TREESORT's expected time is also C*N*log(N); therefore, the quantities in figure 2 also reflect the expected increases for TREESORT.

PROGRAM SORTER

SORTER is a structured BASIC pro-

Eliezer Naddor is a professor in the Department of Mathematical Sciences at The Johns Hopkins University. He is a frequent contributor to this magazine.

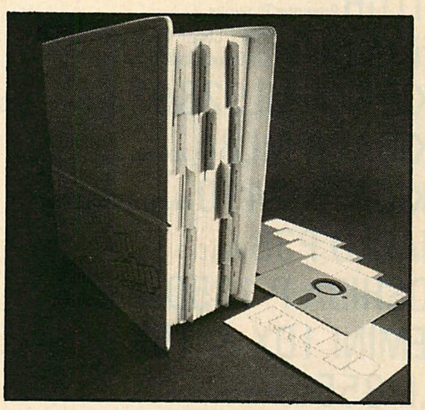
For your IBM/PC

mbp COBOL: 4 times faster, and now with SORT & CHAIN.

$750.

mbp COBOL can be summed up in one word: fast.

Because it generates native machine language object code, the mbp COBOL Compiler executes IBM/PC* programs *at least* 4 times faster (see chart). Fast also describes our **new SORT**, which can sort four-thousand 128-byte records in less than 30 seconds. A callable subroutine or stand-alone, 9 SORT control fields can be specified. And our **new CHAIN** is both fast and secure, conveniently transferring control from one program to another, passing 255 parameters. Plus, **new extensions** to ACCEPT & DISPLAY verbs give better, faster interactive programming.



GIBSON MIX Benchmark Results

Calculated S-Profile
(Representative COBOL statement mix)
Execution time ratio

mbp COBOL	Level II** COBOL	R-M*** COBOL	Microsoft**** COBOL
1.00	4.08	5.98	6.18

128K system with hard disk required. *IBM/PC is an IBM TM; **Level II is a Micro Focus TM; ***A Ryan-McFarland TM; **** A Microsoft TM.

The complete COBOL.

An Interactive Symbolic Debug Package included standard; Multi-Keyed ISAM Structure; listing options allow source & object code, map & cross-reference checking; GSA Certification to ANSI '74 Level II; mbp has it all.

It's no surprise companies like Bechtel, Chase, Citicorp, Connecticut Mutual, and Sikorsky choose mbp COBOL; make it your choice, too. mbp is available at Vanpak Software Centers, or direct; just send the coupon, or call for complete information—today.

mbp COBOL: 4 times faster.

mbp®

mbp Software & Systems Technology, Inc.

7700 Edgewater Drive, Suite 360, Oakland, CA 94621
Phone 415/632-1555

Please send complete mbp COBOL information to:

NAME ____________

COMPANY ____________

ADDRESS ____________

CITY/STATE/ZIP ____________

PHONE ____________

CIRCLE NO. 165 ON READER SERVICE CARD

Figure 1

To	32	64	128	256	512	1024
From						
16	4	16	64	256	1024	4096
32	1	4	16	64	256	1024
64		1	4	16	64	256
128			1	4	16	64
256				1	4	16
512					1	4

Figure 1A

To	32	64	128	256	512	1024
From						
16	2.5	6	14	32	72	160
32	1	2.4	5.6	12.8	28.8	64
64		1	2.33	5.33	12	26.67
128			1	2.29	5.14	11.43
256				1	2.25	5
512					1	2.22

Figure 2

Block	Command	Purpose
100		Preliminaries
300	Dimensions	
400	String constants	
600	Heads and images	
700	Reading of methods, first and last names	
900	Message via M$	
1000	User command	
1200	NEW	New raw items
1300	SORT	Analysis of the sorting command
1700	*HELP*	*Give brief helpful information*
1800	*EXAMPLE*	*Give examples of commands*
1850		*Reading of SORTER.HLP*
2000	*GET X*	*Get items in set X of DATA statements*
2050	*GET file.ext*	*Get items from file.ext*
2070	*SAVE file.ext*	*Save items in file.ext*
2100	*NEW n RANDOM*	*New n random items*
2200	*NEW n INPUT*	*Prompt for n items*
2300	*NEW n FORWARD*	*New n sorted items*
2400	*NEW n REVERSE*	*New n reversed items*

gram, based on a user command language (E. Naddor, "UCL: A User Command Language," *PC Tech Journal*, Vol. 1, No. 3, pp. 44-58). It is suitable for comparing various sorting and searching algorithms and has been used to produce the results in the following section of this article. A copy of the program and supporting documentation may be obtained by writing to the author.

There are four groups of commands in SORTER: 1) information and termination commands, (2) data management commands, (3) a sorting command, and (4) a searching command. Each command starts with a command root and is optionally followed by qualifiers and/or parameters. The roots, the qualifiers, and some of the parameters are English words. SORTER uses only the first three characters of each word for identification.

Before the user is prompted for a command, SORTER prints a panel number, and the current date and time, using the PC's functions DATE$ and TIME$. Because the latter includes seconds, it is possible to compute the number of seconds needed to execute commands in SORTER. (The new BASIC 2.0 function TIMER was not used since it was not recognized by the current BASIC Compiler.)

Information and Termination Commands (HELP, EXAMPLE and STOP). The command HELP gives brief information about SORTER. 'HELP COMMANDS' lists all available commands, 'HELP METHODS' gives the names of the available sorting algorithms, 'HELP REFERENCES' gives references on the algorithms, and 'HELP VERSION' lists the dates on which indicated features were added to SORTER. The results in this article are based on Version 83/07/12.

The command EXAMPLE provides examples of legal commands in SORTER. For example, 'EXAMPLE SORT' lists various ways a user may issue the command SORT. Here are four illustrations:

SORT BUBBLE,QUICK,TREE
SORT/DETAILS INSERT, SHELL
SORT GOOD [=SHELL,QUICK,SING,TREE, HEAP]
SORT ALL

The command STOP terminates the running of SORTER.

Data Management Commands (GET, NEW, DISPLAY, and SAVE). All items used in SORTER are strings of various lengths. Items containing only digits are also treated as strings.

The command GET retrieves a set of raw items for sorting. 'GET A' retrieves a set of nine numbers: 61, 32, 83, 14, 45, 56, 27, 98, and 79. These are used later in the coding and examples section of this article to illustrate our eight sorting algorithms. The command 'GET FILE.EXT' retrieves a set of items

2500		*Initialize sorted items array*
2700	*DISPLAY*	*Display raw or sorted items*
3000	*SORT*	*Main sorting framework*
3100	*SORT SELECTION*	*SELECTION algorithm*
3xxx	*SORT yyy*	*yyy algorithm (BUBBLESORT to TREESORT)*
3900	*SORT HEAPSORT*	*HEAPSORT algorithm*
4000	*SEARCH*	*Search for items*
9600		*Command parsing*
9700		*Panel, date, and time*
9800		*Error routine*
9900		*DATA statements*
9910		*Sorting methods*
9920		*First names*
9930		*Last names*
9940		*Item sets A and B*
999	*STOP*	*Terminate running*

Figure 3

[1] 07-13-1983 22:05:43
Command? NEW/NAM 256 RANDOM

256 random items ready

[2] 07-13-1983 22:06:07
Command? SORT ALL

SELECTION	210 seconds
BUBBLESORT	413 seconds
INSERTION	191 seconds
SHELLSORT	57 seconds
QUICKSORT	40 seconds
SINGLETON	33 seconds
TREESORT	68 seconds
HEAPSORT	59 seconds

[3] 07-13-1983 22:27:43

Figure 3A

Names	32	64	128	256	512	1024
Method						
SELECTION	5	15	56	210	1087	
BUBBLESORT	6	26	98	413	1639	
INSERTION	3	13	45	191	750	
SHELLSORT	4	10	24	57	128	305
QUICKSORT	3	7	17	40	85	179
SINGLETON	3	7	23	33	72	163
TREESORT	7	14	29	68	148	321
HEAPSORT	5	11	26	59	130	291

from the file FILE.EXT. The first line in the file has a header with a title and the number of items. The command 'GET/NOHEAD NAME.ANY' retrieves a set of items from the file NAME.ANY, which does not have a header.

The general format of the command NEW is 'NEW/q1 p1 p2', where: q1 is an optional qualifier, p1 is the first parameter, and p2 is the second. The command produces p1 raw items. If p2 is RANDOM, then the items are produced randomly. If p2 is INPUT, then the user is prompted for p1 items. If p2 is FORWARD, then the items produced are already sorted, and if p2 is REVERSE, then the items are in reverse order. If no qualifier q1 is given, then the random items produced are three-digit strings between 000 and 999. If q1 is NAME, then random items such as 'Martha X. Washington' and 'John Z. Monroe' are produced. The qualifier LAST produces last names (e.g., 'Jefferson') and FIRST produces first names (e.g., 'Rachel'). Note that qualifiers can only be used when p2 is RANDOM.

The command DISPLAY also has an optional qualifier q1, and two parameters p1 and p2, where p2 is optional. The parameters are item numbers and the qualifier is either RAW or SORTED.

Absence of a qualifier is treated as the qualifier RAW. The command 'DISPLAY 5 7' lists the raw items 5, 6, and 7; 'DISPLAY/SORTED 500 512' displays sorted items 500 to 512.

The command SAVE allows the user to save in a designated file either raw items or sorted ones, with or without a header line. 'SAVE 1024.DAT' saves the current raw items with a header in file 1024.DAT; 'SAVE/NOHEAD/SORTED SORT.512' saves the sorted items without a header in file SORT.512.

The Sorting Command (SORT). The general format of the command is 'SORT/q1 p1 p2 . . . pn'. The parameters are names

Figure 1: Expected Increase in Sorting Times for SELECTION, BUBBLESORT, and INSERTION

Figure 1A: Expected Increase in Sorting Times for QUICKSORT, SINGLETON, HEAPSORT, SHELLSORT, and TREESORT

Figure 2: Block Structure of Program SORTER

Figure 3: Sample Inputs and Outputs in SORTER

Figure 3A: Seconds Needed to Sort Names in an Interpreted Program

ANNOUNCING
A PROUD ADDITION
TO YOUR FAMILY.

The next addition to your family could be the bright little newcomer in the growing family of IBM® personal computers.

Name: PC*jr*. Weight: 12 pounds. Heritage: more than 30 years of computer experience.

"Junior" is a powerful tool for modern times. Yet it's simple enough for a child to enjoy.

BRINGING HOME BABY

It's a big day when PC*jr* comes home. The surprises begin the moment you open the carton.

Surprise #1 is the IBM "Freeboard"— a keyboard that doesn't need a connecting cord. The Freeboard frees you to move around and relax.

Then there's the Keyboard Adventure— an instructional exercise for first-time users. It's built into the computer and explained step-by-step in the Guide to Operations. It will help anyone begin learning as soon as PC*jr* is hooked up to a TV set.

In systems equipped with a diskette drive, there's a program that lets you explore computer fundamentals at your own pace, with PC*jr* as your teacher.

And to get you off and running from the very first day, a sample diskette with eleven useful mini-programs (ranging from a spreadsheet for monthly expenses to a word game and a recipe file) is also included.

But there are still more surprises.

FAMILY COMPUTING MADE EASY

Many IBM software programs written for other IBM personal computers will run on PC*jr*. And inexpensive new ones written especially for PC*jr* are being released.

An easy-to-use diskette word processing program, for example, uses pictures as well as words to guide you along. A comprehensive IBM home budget program makes keeping track of money easier. There's also a selection of educational programs for children at home and at school.

And when the work is finished (or perhaps before), the fun can begin. Just slip in a game cartridge and stand back.

GROWING UP WITH JUNIOR

Add a printer. A diskette drive. An internal modem for telecommunications. Increase user memory from 64KB to 128KB. With these and other add-it-yourself options, even the lowest-priced PC*jr* can grow up *real* fast.

PC*jr* is a powerful tool for home, school or college. With its optional carrying case, it's a powerful tool anywhere you care to take it.

SEE JUNIOR RUN

Junior's starting model includes a 64KB cassette/cartridge unit and Freeboard for about $700. A 128KB model with diskette drive is about $1300. (Prices apply at IBM Product Centers. Prices may vary at other stores.)

Your local authorized IBM PC*jr* dealer proudly invites you to see this bright little addition to the family. For the store nearest you, just call 1-800-IBM-PCJR. In Alaska and Hawaii, 1-800-447-0890.

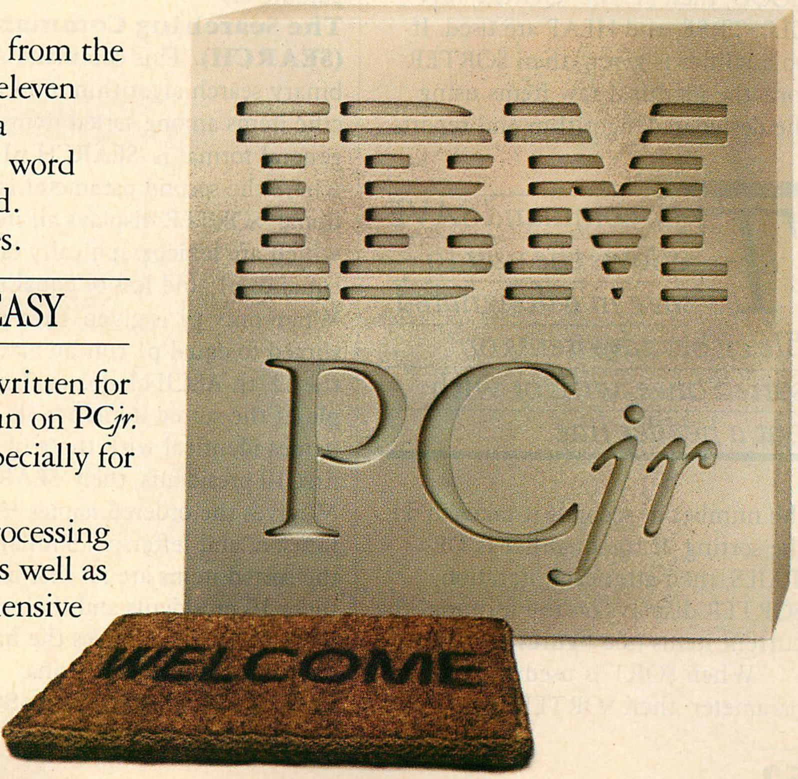


Figure 3B

Names	32	64	128	256	512	1024
Method						
SELECTION	0	0	2	10	37	146
BUBBLESORT	0	0	3	13	50	203
INSERTION	0	0	2	8	30	120
SHELLSORT	0	1	1	2	5	14
QUICKSORT	0	0	1	1	3	9
SINGLETON	0	0	1	2	4	8
TREESORT	1	1	1	1	3	6
HEAPSORT	0	0	2	4	7	13

Figure 3C

Short Items	32	64	128	256	512	1024
Method						
SELECTION	0	1	3	10	36	140
BUBBLESORT	0	1	3	11	47	193
INSERTION	0	0	2	7	29	119
SHELLSORT	0	1	1	2	4	10
QUICKSORT	0	0	1	1	2	5
SINGLETON	0	0	0	2	3	7
TREESORT	0	0	0	1	3	6
HEAPSORT	0	0	2	2	5	12

Figure 3D

Numbers	32	64	128	256	512	1024
Method						
SELECTION	5	17	63	238		
BUBBLESORT	7	25	101	409		
INSERTION	4	14	55	215		
SHELLSORT	4	10	26	63		
QUICKSORT	4	8	18	40		
SINGLETON	3	7	15	34		
TREESORT	6	14	31	67		
HEAPSORT	5	11	26	59		

Figure 3E

Names	32	64	128	256	512	1024
Method						
SELECTION	5	17	63	241		
BUBBLESORT	9	31	123	490		
INSERTION	5	15	56	225		
SHELLSORT	5	11	28	65		
QUICKSORT	4	9	20	48		
SINGLETON	3	8	17	37		
TREESORT	8	15	34	73		
HEAPSORT	6	13	30	66		

Figure 3F

Names	32	64	128	256	512	1024
Method						
SELECTION					36	140
BUBBLESORT					150	598
INSERTION					56	222
SHELLSORT					7	17
QUICKSORT					3	7
SINGLETON					3	7
TREESORT					3	6
HEAPSORT					6	13

Figure 3G

Sorted Items	32	64	128	256	512	999
Method						
SELECTION	0	0	2	9	35	129
BUBBLESORT	0	0	0	0	0	0
INSERTION	0	0	0	0	0	1
SHELLSORT	0	0	1	1	2	6
QUICKSORT	0	0	2	8	32	122
SINGLETON	0	0	0	1	1	3
TREESORT	0	0	0	1	2	4
HEAPSORT	0	0	1	3	5	13

of sorting algorithms (e.g., QUICK, SHELL, and INSERT). If the only parameter given is ALL, then all eight algorithms are used; and if it is GOOD, then SHELL, QUICK, SINGLE, TREE, and HEAP are used. If no qualifier is given, then SORTER sorts the identical raw items using the designated algorithm and reports the number of seconds it takes to do the sorting. If the qualifier is DETAILS, then after each iteration, SORTER displays on the terminal the current items (see figures 4 to 4H).

The command SAVE allows the user to save in a designated file either raw items or sorted ones, with or without a header line.

When SORT is used without any parameter, then SORTER uses the following rule: If SORT is used for the first time, then it assumes that the command is 'SORT GOOD'; otherwise it uses the previous set of parameters.

The Searching Command (SEARCH). This command uses a binary search algorithm to locate specific items among sorted items. Its general format is 'SEARCH p1 p2', where the second parameter is optional. SORTER displays all items which are lexicographically larger or equal to p1, and less or equal to p2. When only p1 is given, then p2 is assumed to equal p1 concatenated with the 127th ASCII character. For example, if the sorted items are all last names identical with those of our first 10 presidents, then 'SEARCH H M' gives the ordered names Harrison, Jackson, and Jefferson. Similarly, if the sorted items are all first names of these 10 presidents and their wives, then 'SEARCH A' gives the names Abigail, Andrew, and Anna.

Structure of Program SORTER. The general structure of SORTER is shown in figure 2. The coding of the sorting framework (block 3000), and that for our eight sorting algorithms (blocks 3100 to 3900) are given in the coding and examples section of this article.

The user is prompted for a command in block 1000. The HELP and EXAMPLE commands are executed in blocks 1700 and 1800; the actual respective text is in a file called SORTER.HLT.

The data management commands are in blocks 2000 (GET), 2100 to 2500 (NEW), 2700 (DISPLAY), and 2070 (SAVE). The sorting command is analyzed in block 1300, while the actual sorting is in blocks 3000 to 3900. The searching command is executed in block 4000.

For additional information the reader is urged to obtain a full listing of the program.

TIMING RESULTS ON THE PC

In figures 3 to 3G we give numerous

Figure 4:

```
3000 FOR MO=1 TO M8:M=M(MO):IO=0
3005 FOR I=1 TO N:V$(I)=W$(I):NEXT I:X=FRE(""):C=FNS(TIME$)
3010 IF P>0 THEN PRINT:PRINT LEFT$(M$(M),10):GOSUB 6900
3015 ON M GOSUB 3100,3200,3300,3400,3500,3600,3800,3900
3020 REM       SEL  BUB  INS  SHE  QUI  SIN  TRE  HEA
3025 T=FNS(TIME$)-C:IF T<0 THEN T=T+3600
3030 PRINT USING I$;M$(M),T:NEXT MO:GOTO 1000

6900 IF P=0 THEN RETURN ELSE PRINT IO;
6905 FOR Q=1 TO N:PRINT " ";V$(Q);:NEXT Q:PRINT:IO=IO+1:RETURN
```

Figure 4A:

```
3100 FOR I=1 TO N-1:T$=V$(I):K=I:FOR J=I+1 TO N
3105 IF V$(J)<T$ THEN T$=V$(J):K=J
3110 NEXT J:IF K>I THEN SWAP V$(K),V$(I)
3115 GOSUB 6900:NEXT I:RETURN

             0:  61 32 83 14 45 56 27 98 79
             1:  14 32 83 61 45 56 27 98 79
             2:  14 27 83 61 45 56 32 98 79
             3:  14 27 32 61 45 56 83 98 79
             4:  14 27 32 45 61 56 83 98 79
             5:  14 27 32 45 56 61 83 98 79
             6:  14 27 32 45 56 61 83 98 79
             7:  14 27 32 45 56 61 79 98 83
             8:  14 27 32 45 56 61 79 83 98
```

Figure 4B:

```
3200 K=N
3205 T=0:FOR I=2 TO K
3210 IF V$(I-1)>V$(I) THEN T=I:SWAP V$(I-1),V$(I)
3215 NEXT I:GOSUB 6900:K=T-1:IF K>1 THEN 3205 ELSE RETURN

             0:  61 32 83 14 45 56 27 98 79
             1:  32 61 14 45 56 27 83 79 98
             2:  32 14 45 56 27 61 79 83 98
             3:  14 32 45 27 56 61 79 83 98
             4:  14 32 27 45 56 61 79 83 98
             5:  14 27 32 45 56 61 79 83 98
             6:  14 27 32 45 56 61 79 83 98
```

Figure 4C:

```
3300 FOR I=2 TO N:T$=V$(I)
3305 FOR J=I-1 TO 1 STEP -1:IF V$(J)<=T$ THEN 3315
3310 SWAP V$(J+1),V$(J):NEXT J:J=0
3315 SWAP V$(J+1),T$:GOSUB 6900:NEXT I:RETURN

             0:  61 32 83 14 45 56 27 98 79
             1:  32 61 83 14 45 56 27 98 79
             2:  32 61 83 14 45 56 27 98 79
             3:  14 32 61 83 45 56 27 98 79
             4:  14 32 45 61 83 56 27 98 79
             5:  14 32 45 56 61 83 27 98 79
             6:  14 27 32 45 56 61 83 98 79
             7:  14 27 32 45 56 61 83 98 79
             8:  14 27 32 45 56 61 79 83 98
```

results obtained by running SORTER and using our eight sorting methods on various sets of items. The following methodology is observed: A set of random items is generated with the command NEW, and the items are placed in the array W$(). For each sorting algorithm, the array W$() is copied into an array V$(), and the time is then noted using TIME$. The array V$() is then sorted, and when sorting is complete the time is again noted with TIME$. The difference in seconds between the two times is then printed.

Figure 3 shows an example using this procedure. It took SORTER 24 seconds to generate 256 names (from 22:05:43 to 22:06:07). To sort the names with eight algorithms, SORTER took 21 minutes and 36 seconds (from 22:06:07 to 22:27:43). This time is somewhat larger than the sum of the individual times given for the sorting (1296 seconds as compared to 1071). This time difference was used by SORTER for a variety of bookkeeping chores. More will be said about this later.

If the parameter given is ALL, all 8 algorithms are used; if it is GOOD, SHELL, QUICK, SINGLE, TREE, and HEAP are used.

SORTER in Interpreted Mode. Figure 3A gives the sorting times when SORTER is used in the interpreted mode of BASIC. Note that the times of figure 3 have been correctly entered in column 256 of figure 3A. A close examination of the times for SELECTION, BUBBLESORT, and INSERTION will confirm that they follow the theoretically expected times of CN^2, which are discussed above in Knuth's Analysis of Sorting Algorithms. Similarly, the times for the other methods correspond to C*N*log(N).

Figure 3B: Seconds Needed to Sort Names in a Compiled Program

Figure 3C: Seconds Needed to Sort Short Items in a Compiled Program

Figure 3D: Seconds Needed to Sort Numbers in an Interpreted Program

Figure 3E: Seconds Needed to Sort Names in an Interpreted Program with Non-integer Variables

Figure 3F: Seconds Needed to Sort Names in a Compiled Program without SWAP Statements

Figure 3G: Seconds Needed to Sort Sorted Items in a Compiled Program

Figure 4: Sorting Framework

Figure 4A: The SELECTION Algorithm

Figure 4B: The BUBBLESORT Algorithm

Figure 4C: The INSERTION Algorithm

Figure 4D

```
3400 D=INT((3^INT(LOG(2*N+.1)/LOG(3)))/2)
3405 FOR I=D+1 TO N:T$=V$(I)
3410 FOR J=I-D TO 1 STEP -D:IF V$(J)<=T$ THEN 3420
3415 SWAP V$(J+D),V$(J):NEXT J:IF J>0 THEN J=J-D
3420 SWAP V$(J+D),T$:GOSUB 6900:NEXT I
3425 D=INT(D/3):IF D>=1 THEN 3405 ELSE RETURN

 0:  61 32 83 14 45 56 27 98 79
 1:  45 32 83 14 61 56 27 98 79
 2:  45 32 83 14 61 56 27 98 79
 3:  45 32 27 14 61 56 83 98 79
 4:  45 32 27 14 61 56 83 98 79
 5:  45 32 27 14 61 56 83 98 79
 6:  32 45 27 14 61 56 83 98 79
 7:  27 32 45 14 61 56 83 98 79
 8:  14 27 32 45 61 56 83 98 79
 9:  14 27 32 45 61 56 83 98 79
10:  14 27 32 45 56 61 83 98 79
11:  14 27 32 45 56 61 83 98 79
12:  14 27 32 45 56 61 83 98 79
13:  14 27 32 45 56 61 79 83 98
```

Figure 4F

```
3600 L=1:H=N:S=0:V$(0)="":V$(N+1)=Z9$:GOTO 3622
3602 J=J-1
3604 IF V$<V$(J) THEN 3602 ELSE IF I>=J THEN 3612 ELSE V$(I)=V$(J)
3606 I=I+1
3608 IF V$(I)<V$ THEN 3606
3610 IF I<J THEN V$(J)=V$(I):GOTO 3602 ELSE I=J
3612 SWAP V$(I),V$:S=S+1:GOSUB 6900
3614 IF I-L<H-I THEN B(S)=I+1:E(S)=H:H=I-1 ELSE B(S)=L:E(S)=I-1:L=I+1
3616 IF L=H-2 THEN 3636 ELSE IF L<H-2 THEN 3622
3618 IF V$(L)>V$(L+1) THEN SWAP V$(L),V$(L+1)
3620 GOSUB 6900:L=B(S):H=E(S):S=S-1:IF S>=0 THEN 3616 ELSE RETURN
3622 I=L:J=H:K=(I+J)/2:X$=V$(I):Y$=V$(K):Z$=V$(J)
3624 IF X$>=Y$ THEN 3628 ELSE IF X$>=Z$ THEN 3634
3626 IF Y$<Z$ THEN 3632 ELSE GOTO 3630
3628 IF X$>Z$ THEN IF Y$<=Z$ THEN 3630 ELSE GOTO 3632 ELSE GOTO 3634
3630 V$=Z$:GOTO 3608
3632 V$=Y$:V$(K)=Z$:GOTO 3608
3634 V$=X$:GOTO 3604
3636 K=L+1:X$=V$(L):Y$=V$(K):Z$=V$(H)
3638 IF X$>Y$ THEN 3644 ELSE IF Y$<=Z$ THEN 3620
3640 V$(H)=Y$:IF X$>Z$ THEN V$(L)=Z$:V$(K)=X$:GOTO 3620
3642 V$(K)=Z$:GOTO 3620
3644 IF X$<=Z$ THEN V$(L)=Y$:V$(K)=X$:GOTO 3620
3646 V$(H)=X$:IF Y$<Z$ THEN V$(L)=Y$:V$(K)=Z$:GOTO 3620
3648 V$(L)=Z$:GOTO 3620

0:  61 32 83 14 45 56 27 98 79
1:  27 32 56 14 45 61 83 98 79
2:  27 32 56 14 45 61 79 83 98
3:  27 32 14 45 56 61 79 83 98
4:  27 32 14 45 56 61 79 83 98
5:  14 27 32 45 56 61 79 83 98
```

Figure 4E

```
3500 L=1:H=N:S=0:V$(0)="":V$(N+1)=Z9$
3505 I=L:J=H+1:V$=V$(I)
3510 J=J-1:IF V$<V$(J) THEN 3510
3515 IF I>=J THEN 3530 ELSE V$(I)=V$(J)
3520 I=I+1:IF V$(I)<V$ THEN 3520
3525 IF I<J THEN V$(J)=V$(I):GOTO 3510 ELSE I=J
3530 SWAP V$(I),V$:S=S+1:GOSUB 6900
3535 IF I-L<H-I THEN B(S)=I+1:E(S)=H:H=I-1 ELSE B(S)=L:E(S)=I-1:L=I+1
3540 IF L<H THEN 3505 ELSE L=B(S):H=E(S):S=S-1
3545 IF S>=0 THEN 3540 ELSE RETURN

0:  61 32 83 14 45 56 27 98 79
1:  27 32 56 14 45 61 83 98 79
2:  27 32 56 14 45 61 79 83 98
3:  14 27 56 32 45 61 79 83 98
4:  14 27 45 32 56 61 79 83 98
5:  14 27 32 45 56 61 79 83 98
```

The times of figure 3A represent one set of computations with SORTER. Several other sets have been run—the results were always within a few seconds of those in the figure.

SORTER in Compiled Mode. SORTER was compiled with version 1.00 of IBM's and Microsoft's compiler. Running SORTER in this mode produced the dramatic results of figure 3B. I expected the compiled program to run about 10 times faster than the interpreted one. But figure 3B (as compared with figure 3A) shows that the sorting times have been reduced in many cases by factors of 30 to 50.

The greatest surprise was TREESORT. In the interpreted mode it was slower than all other 'good' algorithms, but when it is compiled it seems to be the fastest sorting method.

Short vs. Long Items. Each item used in figures 3A and 3B was a name composed of a first name, a middle initial and a last name. SORTER uses 17 first names (from short ones as Anna and John to a long one, Elizabeth). When generating, say, 512 names, one can expect that about every 30 words would start with the same first name. I naturally expected that the time to sort such names will be longer than sorting shorter items, such as 3-digit strings.

In figure 3C are the times for sorting these shorter items. Surprisingly, if we compare the times with the results of figure 3B, we find that the length of items has only a minute effect on sorting times.

Sorting of Numbers. In order to check sorting times of numbers, as compared with the sorting times of strings, SORTER was rewritten. In the new version of SORTER, the items were 3-digit integers, which were initially placed in W() and then sorted in V().

The results are in figure 3D, and are based on the interpreted mode of BASIC. Comparing them with the results of figure 3A, we note that the times are about the same. It appears, then, that on the PC it takes about the same time to sort integers as it takes to sort strings.

The Effect of Integer Variables. Line 110 in SORTER has the statements DEFINT A-W and DEFSNG X-Z. The variables in all sorting algorithms are integers. Only TREESORT uses a few single precision variables. If line 110 is removed from SORTER, all variables are then treated as single precision variables.

The results in figure 3E reflect the effect of the change. As expected, the sorting times are larger than the corresponding ones in figure 3A.

The Effect of the SWAP Statement. Many readers are probably not aware that the SWAP statement may sometimes have a great effect on the performance of a program. My son, David, drew my attention to this, and I checked it with SORTER.

Initially, SORTER was written without the SWAP statement. Figure 3F gives the sorting times in a com-

Figure 4G

```
3800 FOR I=1 TO N+T9:A(I)=0:NEXT I:Y1=0:I=0:M1=0:Z2=0:Z4=0:J=N+1
3802 A(1)=1:A(J)=1:Y2=1:IF N <= 1 THEN 3846 ELSE Z0=N
3804 IF Z0<4 THEN 3808 ELSE Y2=2*Y2:X2=Z0/2
3806 Z0=INT(X2):Z4=Z4+(X2-Z0)*Y2:GOTO 3804
3808 Z4=Y2-Z4:X2=Y2/2
3810 IF Y1=Y2 THEN 3846 ELSE Y1=Y1+1:Z1=Y1:X1=X2:Z3=Z2
3812 Z1=Z1/2:IF INT(Z1)>=Z1 THEN M1=M1+1:Z2=Z2-X1:X1=X1/2:GOTO 3812
3814 Z2=Z2+X1:IF Z0=2 THEN 3820
3816 IF Z3<Z4 THEN 3822
3818 M1=-M1:GOTO 3826
3820 IF Z3<Z4 THEN 3824
3822 M1=M1+1:I=I+1:A(I)=i:A(J)=I:J=J+1
3824 M1=M1+1
3826 I=I+1:L1=I:A(I)=I:A(J)=I:L0=J:J=J+1
3828 I=I+1:L2=I:A(I)=I:A(J)=I:GOTO 3832
3830 J=J-1:L0=J-1:L1=A(L0):L2=A(J)
3832 IF W$(L1)<=W$(L2) THEN 3838 ELSE A(L0)=L2
3834 L0=L2:L2=A(L0):IF L2=L0 THEN 3840
3836 IF W$(L1)>W$(L2) THEN 3834 ELSE A(L0)=L1
3838 L0=L1:L1=A(L0):IF L1<>L0 THEN 3832 ELSE A(L0)=L2:GOTO 3842
3840 A(L0)=L1
3842 M1=M1-1:IF M1>0 THEN 3830 ELSE IF M1=0 THEN GOSUB 3850:GOTO 3810
3844 M1=1-M1:GOTO 3826
3846 L0=N+1:FOR I=1 TO N:L0=A(L0):V$(I)=W$(L0):NEXT I:RETURN

3850 IF P=0 THEN RETURN ELSE PRINT I0;:K=N+1:K0=0
3852 FOR Q=1 TO N:K=A(K):IF K=K0 THEN L=Q ELSE L=K
3854 PRINT" ";W$(L);:K0=K:NEXT Q:PRINT:I0=I0+1:RETURN

                  0:  61 32 83 14 45 56 27 98 79
                  1:  32 61 83 14 45 56 27 98 79
                  2:  14 32 61 83 45 56 27 98 79
                  3:  14 32 61 83 45 56 27 98 79
                  4:  14 27 32 45 56 61 79 83 98
```

Figure 4H

```
3900 R=N:FOR L=INT(N/2) TO 1 STEP -1:T$=V$(L):GOSUB 3950:NEXT L:L=1
3905 FOR R=N-1 TO 1 STEP -1:T$=V$(R+1):V$(R+1)=V$(1):GOSUB 3950:NEXT R
3910 RETURN

3950 J=L
3955 I=J:J=2*J:ON 2+SGN(J-R) GOTO 3960,3965,3970
3960 IF V$(J)<V$(J+1) THEN J=J+1
3965 IF T$<V$(J) THEN V$(I)=V$(J):GOTO 3955
3970 SWAP V$(I),T$:GOSUB 6900:RETURN

                  0:  61 32 83 14 45 56 27 98 79
                  1:  61 32 83 98 45 56 27 14 79
                  2:  61 32 83 98 45 56 27 14 79
                  3:  61 98 83 79 45 56 27 14 32
                  4:  98 79 83 61 45 56 27 14 32
                  5:  83 79 56 61 45 32 27 14 98
                  6:  79 61 56 14 45 32 27 83 98
                  7:  61 45 56 14 27 32 79 83 98
                  8:  56 45 32 14 27 61 79 83 98
                  9:  45 27 32 14 56 61 79 83 98
                 10:  32 27 14 45 56 61 79 83 98
                 11:  27 14 32 45 56 61 79 83 98
                 12:  14 27 32 45 56 61 79 83 98
```

piled program. Then, SWAP was used in all algorithms. For example, in BUBBLESORT the three statements T$=V$(I), V$(I)=V$(I-1), and V$(I-1)=T$, were replaced by SWAP V$(I),V$(I-1). Also, when appropriate, statements of the form A$=B$ were replaced by SWAP A$,B$. This led to the version of SORTER used to produce the results in figure 3A to 3E.

In comparing figures 3B and 3F, the greatest effect of the SWAP statement is noticed for BUBBLESORT and INSERTION.

In order to check sorting times of numbers, as compared with the sorting times of strings, SORTER was rewritten.

Sorting of Sorted Names. In reporting on Knuth's analysis, I failed to emphasize that his results pertained to random sets of items. The expected increases of figures 1 and 1A do not hold when the items in a set are already sorted. This is particularly true, in a contrasting way, for QUICKSORT and BUBBLESORT.

In figure 3G we give the sorting times for items that are already sorted. (The raw items were obtained with the command 'NEW xx FORWARD', where xx is the desired number of items.)

Our good algorithm QUICKSORT is not so good here. This is the principal reason for the SINGLETON algorithm, which is based on QUICKSORT, and handles well both sorted and random items.

The reader will notice that instead of 1024 items, we have only 999 in figure 3G. When 1024 forward items were generated and sorted, the respective times were 139, 151, 12, 7, 11, 8, 5, and 13. I knew immediately that something was wrong. Before reading further, can you find the reason for these results?

Figure 4D: The SHELLSORT Algorithm

Figure 4E: The QUICKSORT Algorithm

Figure 4F: The SINGLETON Algorithm

Figure 4G: The TREESORT Algorithm

Figure 4H: The HEAPSORT Algorithm

Figure 5

Names	32	64	128	256	512	1024
Method						
SELECTION	0.0	0.0	0.3	1.3		
BUBBLESORT	0.0	0.0	1.3	5.1		
INSERTION	0.0	0.0	0.4	2.0		
SHELLSORT	0.0	0.0	0.2	0.4	0.9	2.2
QUICKSORT	0.0	0.0	0.1	0.1	0.4	0.8
SINGLETON	0.0	0.0	0.0	0.2	0.4	0.8
TREESORT	0.0	0.0	0.0	0.1	0.3	0.6
HEAPSORT	0.0	0.0	0.2	0.3	0.7	1.4

Figure 5B

Numbers	32	64	128	256	512	1024
Method						
SELECTION				0.6	2.6	
BUBBLESORT				1.6	6.5	
INSERTION				0.7	2.5	
SHELLSORT				0.2	0.4	0.8
QUICKSORT				0.1	0.2	0.4
SINGLETON				0.1	0.2	0.3
TREESORT				0.1	0.3	0.5
HEAPSORT				0.2	0.4	1.0

Figure 5A

Short Items	32	64	128	256	512	1024
Method						
SELECTION	0.1	0.1	0.3	1.1		
BUBBLESORT	0.0	0.3	1.1	5.0		
INSERTION	0.0	0.1	0.5	1.8		
SHELLSORT	0.0	0.1	0.2	0.4	1.0	2.2
QUICKSORT	0.1	0.1	0.1	0.2	0.3	0.8
SINGLETON	0.0	0.0	0.0	0.2	0.4	0.8
TREESORT	0.0	0.0	0.1	0.1	0.3	0.7
HEAPSORT	0.0	0.1	0.2	0.4	0.8	1.8

Figure 5C

Sorted Items	32	64	128	256	512	999
Method						
SELECTION	0.0	0.1	0.4	1.2	4.3	16.7
BUBBLESORT	0.0	0.0	0.0	0.1	0.1	0.0
INSERTION	0.0	0.0	0.0	0.0	0.2	0.3
SHELLSORT	0.1	0.1	0.1	0.2	0.7	1.1
QUICKSORT	0.0	0.1	0.2	0.9	3.5	14.1
SINGLETON	0.0	0.0	0.0	0.1	0.2	0.4
TREESORT	0 0	0.0	0.0	0.1	0.2	0.4
HEAPSORT	0.0	0.1	0.2	0.4	1.0	1.8

Recall that the command NEW, with no qualifier, generates 3-digit items. For the FORWARD parameter, the items are 001, 002, etc. The 999th item is 999, the 1000th item is 000, and the last item is 024. These items are not sorted and the results for 1024 items are indeed correct. By limiting the items to 999, a sorted set was achieved and the results in figure 3G were then obtained.

The Effect of Garbage Collection. When early versions of SORTER were run on the PC, the reported sorting times were frequently much larger than expected. It was obvious that this was due to the PC's housecleaning of strings, or what is commonly called garbage collection. In order to get a more accurate time for sorting alone, the statement X=FRE("") was placed after the items were copied from W$() to V$() and just before sorting was started (see line 3005 in figure 4).

For large sets of items another garbage collection still occurs occasionally. The reader should be aware of this phenomenon.

Using the SORT Filter. As a final use of SORTER, I compared its compiled sorting times with those of the SORT filter in DOS 2.0. There was no direct way to find how long it takes the filter to sort items in a file. I, therefore, used the following approach: 1024 random items were generated by SORTER and saved without a header in a file called 1024.RAW. Then the commands 'GET/NOH 1024.RAW', 'SORT TREE' and 'SAV/NOH/SOR 1024.SOR' were issued. This took a total of 46 seconds (GET—16 seconds, SORT—13, of which only 6 were for the actual sorting, and SAVE—17 seconds). Then I gave the DOS commands 'TIME', 'SORT <1024.RAW >1024.SSS', and 'TIME'. And this gave a total of 87 seconds. The tentative conclusion is that the TREESORT algorithm (and each of our other 'good' algorithms) seems to be faster than the algorithm used by the SORT filter.

Figure 5: Seconds Needed to Sort Names in a Compiled Program

Figure 5A: Seconds Needed to Sort Short Items in a Compiled Program

Figure 5B: Seconds Needed to Sort Numbers in a Compiled Program

Figure 5C: Seconds Needed to Sort Sorted Items in a Compiled Program

Coding and Examples of Eight Sorting Algorithms

The coding of our sorting algorithms is given in figures 4A and 4H. For each algorithm we also give an example of the sorting process using nine items. The examples were obtained from SORTER with the command 'SORT/DETAILS'.

In all algorithms there are N raw items in W$(). The actual sorting is done using the array V$(). All other variables are local in the respective subroutines.

The sorting framework is in figure 4. Here, M8 is the number of desired algorithms, and M(M0) is the number of the M0th algorithm. M8 and M() are determined in block 1300 (not shown). P is a flag: P=0 implies no details are to be printed, and P>0 implies the printing of details. P is also determined in 1300.

Subroutine 6900 prints details when P>0. It is first called in line 3010 and is also called by each sorting algorithm (except TREESORT, which has its own printing routine).

The SELECTION algorithm is in figure 4A. K is the location of the smallest item. BUBBLESORT is in figure 4B. Note the use of the SWAP statement in line 3210.

The INSERTION and SHELLSORT algorithms are in figures 4C and 4D. Note how the second algorithm is obtained from the first. The function in 3400 (due to von Saleski) gives the largest D in the series 1, 4, 13, 40, . . . , which is smaller or equal to N.

QUICKSORT and SINGLETON are in figures 4E and 4F. Both need the arrays B() and E(), which hold at most about $\log_2 N$ elements (Knuth, p. 116). They also need Z9$=CHR$(127). The coding in lines 3622-3634 finds Singleton's median. Lines 3636-3648 sort three items, while line 3618 sorts two. SINGLETON's coding is my adaptation of von Saleski's.

TREESORT is in figure 4G. Its printing routine of lines 3850-3854, which is called only from 3842, should be omitted if details are not needed. The coding here is also my adaptation of von Saleski's original coding. Note that TREESORT uses the array A(), which must be dimensioned for $N+2+\log_2 N$ items.

The HEAPSORT algorithm is in figure 4H. Note the two phases in lines 3900 and 3905; each calls the subroutine in 3950.

Readers are requested to draw my attention to possible improvements in the coding of the algorithms.

Sorting on the VAX/VMS

A program identical to SORTER was implemented on DEC's VAX minicomputer, with the VMS operating system, using VAX BASIC.

Figures 5 to 5C give the sorting times for our eight sorting algorithms. (SORTER on the VAX has a ninth algorithm in which QUICKSORT is implemented using a recursive function definition. The times for this algorithm are somewhat larger than for QUICKSORT.)

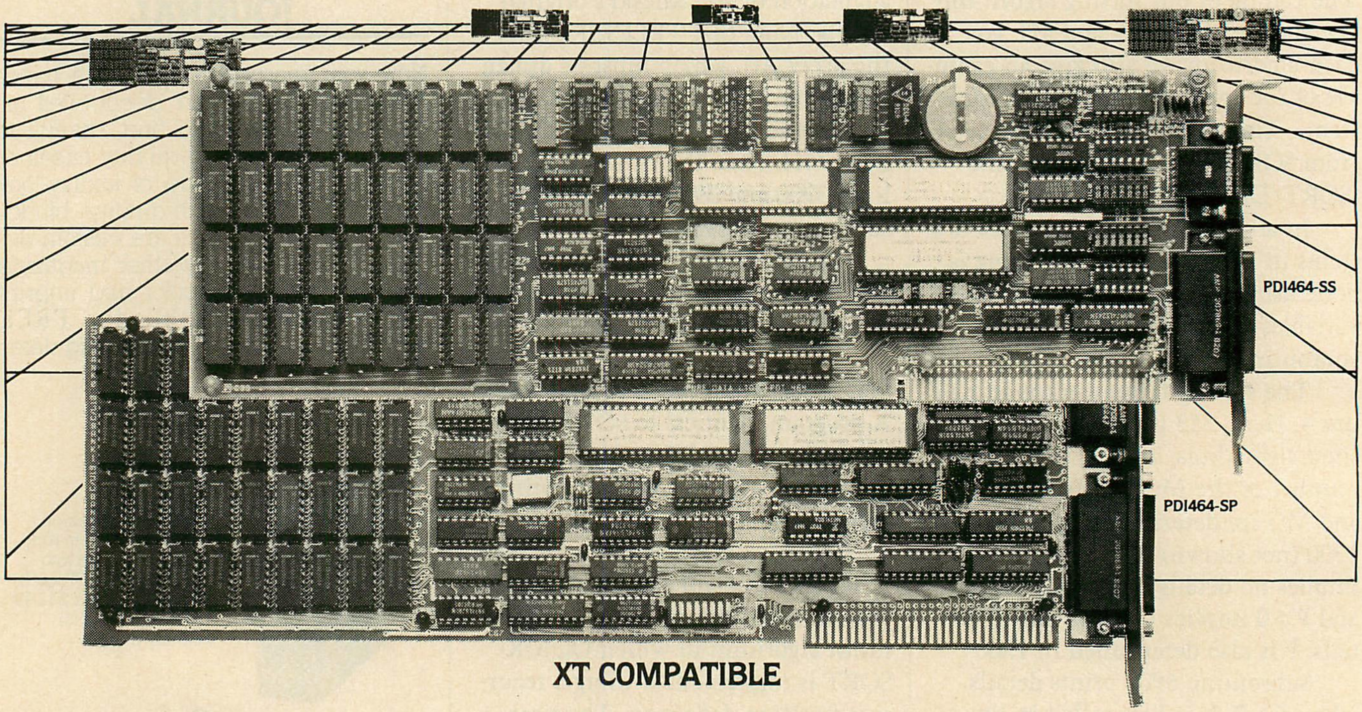
Comparing figures 5 and 3B, in which SORTER was run in compiled mode, we see that the VAX sorts about 10 times faster than the PC. We also note, once again, that BUBBLESORT is the slowest and TREESORT the fastest.

Sorting of short and long items takes about the same time (figures 5 and 5A). This was also the case on the PC. However, sorting of numbers on the VAX (figure 5B) seems to be faster than sorting of strings—a result different from the one on the PC.

Finally, in figure 5C are the times for sorting sorted items. As expected, QUICKSORT performs poorly, SINGLETON does superbly, and so does TREESORT.

Copies of the VAX version of program SORTER and supporting documentation may also be obtained from the author.

Plug Pure Data



XT COMPATIBLE

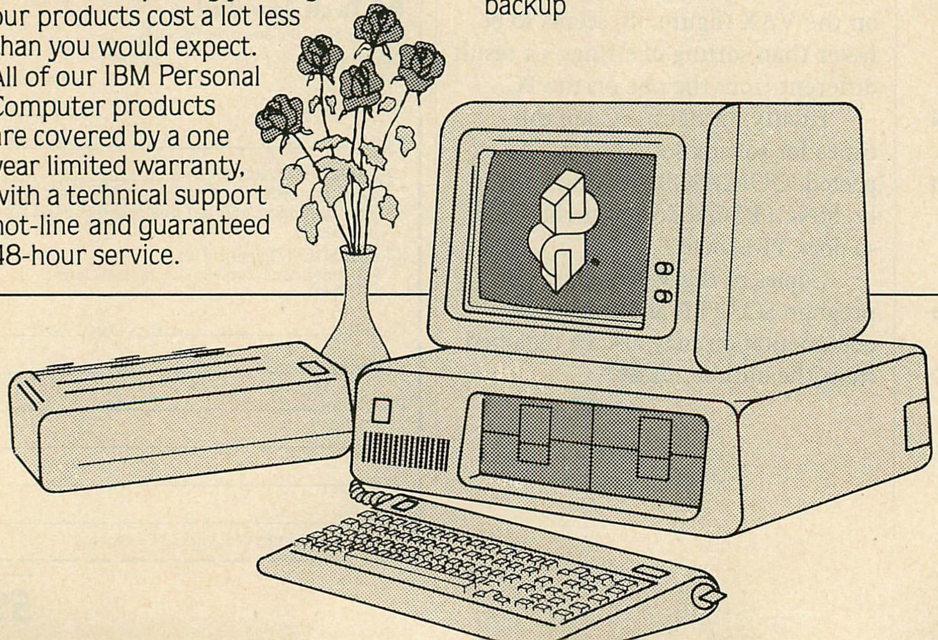
If you're looking for a multi-function board, you're faced with a bewildering array of functions, manufacturers, and prices. It's important to select the right card for your present and future needs. Pure Data offers a variety of high-quality expansion boards designed for the utmost in reliability and flexibility. We invite you to make comparisons with other brands. Our support software and documentation is the most complete available. Surprisingly enough, our products cost a lot less than you would expect. All of our IBM Personal Computer products are covered by a one year limited warranty, with a technical support hot-line and guaranteed 48-hour service.

PDI464-SP

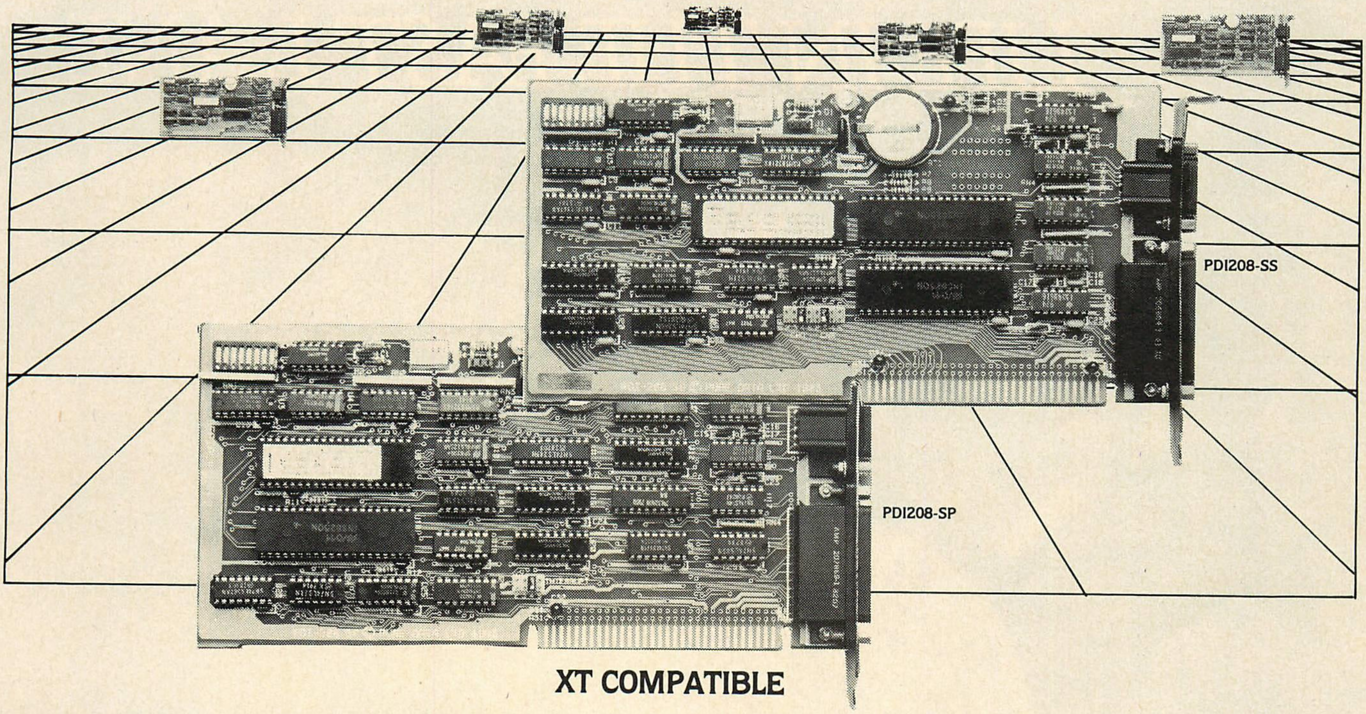
- Up to 256K of RAM, fully socketed, with parity generation/checking
- Highest quality ceramic RAM chips used throughout
- IBM-compatible serial channel with selectable address and 6-foot cable
- IBM-compatible parallel printer adaptor with selectable address
- Real Time Clock/Calendar with field-replaceable lithium battery backup
- p-Disk™ disk emulator for single/double-sided diskette emulation
- Print spoolers for parallel and serial channels with multiple copies
- Software for clock support, dynamic memory configuration, memory testing
- Only occupies one expansion slot
- Fully-illustrated installation and operation manual

PDI464-SS

- Up to 256K of RAM, fully socketed, with parity generation/checking
- Highest quality ceramic RAM chips used throughout
- Two IBM-compatible serial channels with selectable addresses, 6-foot cables



into your PC!



XT COMPATIBLE

- Serial channels feature modem/printer configuration jumpers
- Real Time Clock/Calendar with field-replaceable lithium battery backup
- p-Disk™ disk emulator for single/double-sided diskette emulation
- Print spoolers for parallel and serial channels with multiple copies
- Software for clock support, dynamic memory configuration, memory testing
- Only occupies one expansion slot
- Fully-illustrated installation and operation manual

PDI208-SP

- IBM-compatible serial channel with selectable address and 6-foot cable
- IBM-compatible parallel printer adaptor with selectable address
- Real Time Clock/Calendar with field-replaceable lithium battery backup
- p-Disk™ disk emulator for single/double-sided diskette emulation
- Print spoolers for parallel and serial channels with multiple copies
- Software for clock support, dynamic memory configuration, memory testing
- Only occupies one expansion slot
- Fully-illustrated installation and operation manual

PDI208-SS

- Two IBM-compatible serial channels with selectable addresses, 6-foot cables
- Real Time Clock/Calendar with field-replaceable lithium battery backup
- p-Disk™ disk emulator for single/double-sided diskette emulation
- Print spoolers for parallel and serial channels with multiple copies
- Software for clock support, dynamic memory configuration, memory testing
- Only occupies one expansion slot
- Fully-illustrated installation and operation manual

Pure Data Ltd

950 Denison Street, #17,
Markham, Ontario, Canada L3R 3K5
☎ (416) 498-1616 ☎ (416) 475-2424

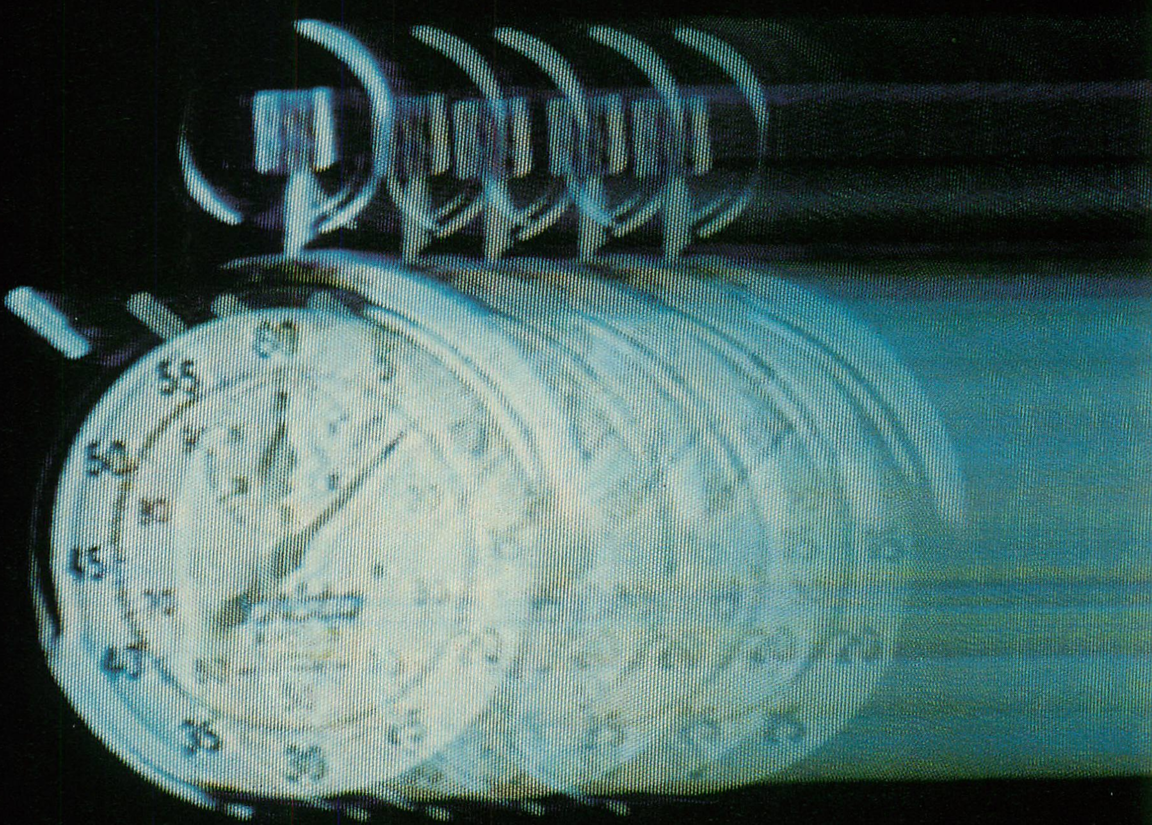
Pure Data products are available through.

OnTrak DATA SYSTEMS 1335 Valwood Parkway, Suite 108,
Carrollton, Texas 75006
Telephone (214) 620-8000
P.O. Box 815155, Dallas, Texas 75381

Keysoft International, Computerland, Computer Innovations, Compugroup, ECOSEA Technologies and others.

CIRCLE NO. 190 ON READER SERVICE CARD

LIFE IN THE FAST LANE



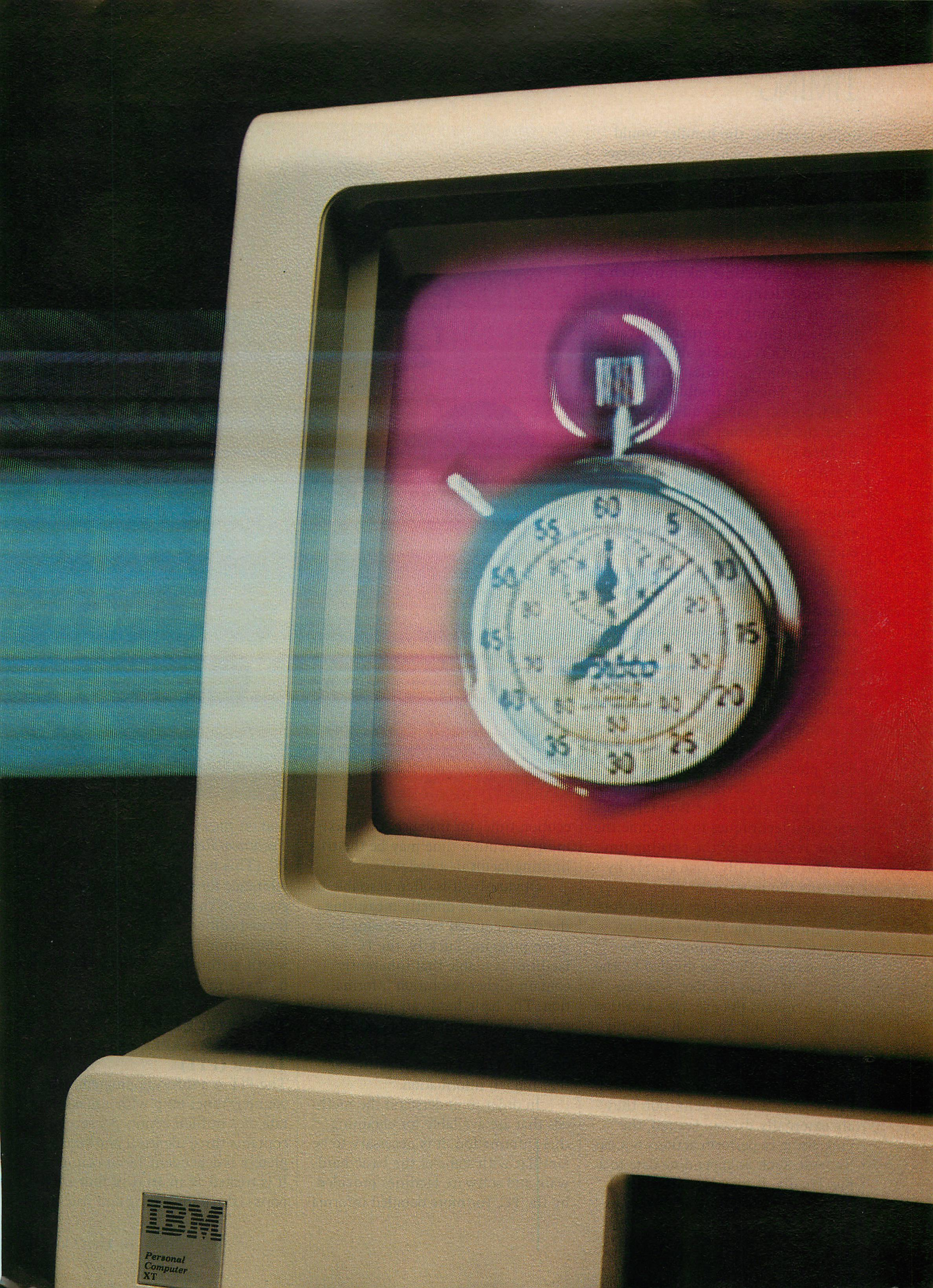
Techniques for obtaining timing information with microsecond resolution on the PC

BOB SMITH and TOM PUCKETT

Accurate timing information is critical to the success of many operations using the personal computer. The standard timing information available to IBM PC users is obtained by invoking DOS function call 2Ch. This returns the time of day in hours, minutes, seconds, and hundredths of a second. Although this resolution is entirely satisfactory for activities measured in human time scales, it is close to useless for events measured on electronic time scales.

Consider the problem of designing an interrupt handler for data arriving on the asynchronous communications port of the PC. At a data-transmission rate of 1200 bits per second, a character can arrive every 8.3 milliseconds. This means that a useful interrupt handler must be able to process each received character (typically by placing it in a circular queue for later processing) in less than 8.3 milliseconds. Prac-

PHOTOGRAPH • MARC COHEN


IBM
Personal
Computer
XT

tically speaking, the handler would need to process each received character in much less than this amount of time. At higher transmission rates, the allowable processing time becomes proportionately less. It therefore becomes a critical matter to correctly assess the time taken by such a handler. Attempts to determine this time by running the handler and relying on timing information available from DOS quickly show that the 10-millisecond resolution will not be adequate. The true resolution is much coarser than 10 milliseconds, since the DOS timer information is updated only every 55 milliseconds.

Another approach to obtaining accurate timing information is to examine the instructions that will be executed and use the standard reference information on the 8088 processor to calculate the time. In practice there are two difficulties with this approach. One is that the flow of control through a complicated handler may be difficult to determine, so it is not easy to be sure that the correct sequence of instructions is being considered. The other is that the reference information cannot evaluate the effect of wait cycles introduced in the 8088 processing by external storage and peripheral devices.

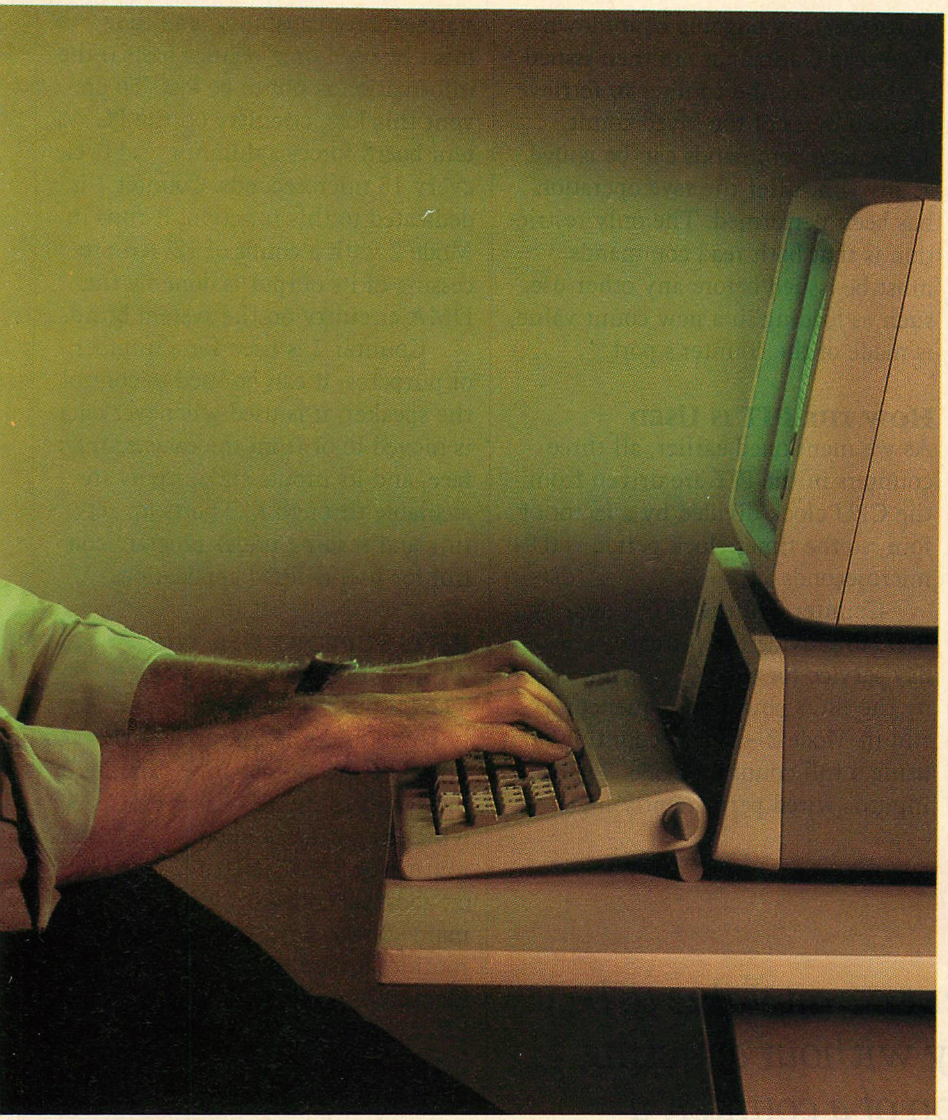
A third alternative is to rely on statistical techniques to overcome the coarse resolution of the existing timer. This is typically done simply by running the program of interest over and over again in a loop until its total accumulated time is large with respect to the timer resolution. The time used can then be divided by the number of times the loop was executed to obtain the average execution time. While this approach is useful for some programs, there are many for which it fails. First, it is generally difficult to put interrupt handlers into a loop, since by definition they are externally driven. Second, the program may perform actions not easily reset, such as appending data to a file. Finally, it may be critical to know more than just the average execution time; for instance, the maximum execution time may be the true limiting factor in a design.

Obviously, it is often desirable to have a means of directly measuring the time required for processing by a given program. Luckily, the PC has facilities available that inherently contain higher-resolution information. The basic hardware timer used to provide all timing information actually runs at better than a microsecond rate. The problem is getting at that hidden information.

In order to understand the methods that are available for obtaining that information, it is necessary to be familiar with some of the basic hardware and software facilities provided by the PC. For more detailed information on these facilities, see the IBM *Technical Reference* manual for the PC. (This manual is an invaluable reference for serious study of what goes on in the PC at this level. The revised edition contains much more information than the first and is worth getting even if you already have the first edition.)

THE INTEL 8253-5 PROGRAMMABLE INTERVAL TIMER (PIT)

The PIT is a software-programmable counter/timer chip with a maximum rate of 2 million counts per second. It contains three identical and completely independent 16-bit counters. It is treated as an array of four I/O ports—three for setting and reading



each of the three counters and the fourth for control purposes. The PIT is driven by the PC's CPU clock divided by four, so its clock rate is 1.193182 MHz, and its clock period is 0.838095 microseconds, which we will round off to 0.84. Because the count value and mode of operation of each counter are under software control, the counters can have many uses. Typical uses are as a programmable rate generator, event counter, binary rate multiplier, real-time clock, and digital one-shot. (See the *Peripheral Design Handbook*, which is published by Intel.)

A given counter must first be initialized to define the mode of operation it will use and then loaded with the starting count value. The count is always downwards from the starting value, and generally an output is available to signal when the count reaches zero. In addition, the capability for reading the current value of a counter at any time is provided. A counter may use either binary or binary coded decimal values in its operation. Besides the clock input, each counter has a second control input (called the gate) that can be used to modify the counter's operation in various ways, depending upon the mode of operation in use.

A counter is controlled by writing a byte to the control port of the PIT that defines which counter is to be affected and whether a counter set or read action is to follow. For a set action, the operation mode and counting mode (binary or BCD) are also defined in the initial write to the control port. The initial write is followed by two writes, this time against the port for the selected counter, which define the starting count value to be used. For a read action no modes need be defined, and the next two reads against the port for the selected counter will return the current residual count value (i.e., the number of counts remaining until the current cycle reaches zero).

There are six counting modes available. The electrical signals representing the gate input and counter output are logic levels, which means that they are normally at either a low- or a high-voltage level. When a level is read or set under program control through the I/O ports, it is represented in the data processed by the program as a bit with a value of either zero or one.

Mode 0 provides an interrupt at the end of the count function. The counter output goes low when the mode is set, and the countdown starts when the count value is subsequently loaded. The output goes high when the count reaches zero. Normally, the gate input is high, but it may be lowered to temporarily suspend the countdown at any point. After the output goes high it stays there until the counter is reloaded.

In Mode 1 the counter acts as a programmable one-shot device. In this case the output goes low and the countdown begins when the gate input rises, and the output goes high when the count reaches zero. The countdown process is started again if the gate signal is lowered and raised another time.

In Mode 2 the counter acts as a rate generator or a divide-by-*n* counter. The output is normally high. When the gate signal goes high the countdown starts from the initial value, which is a convenient way to synchronize the counter cycle with a program event. The counter output drops low for only one timer-clock cycle when zero is reached. The cy-

PHOTOGRAPH • MARC COHEN

cle then repeats automatically.

In Mode 3 the counter runs in a way similar to that of Mode 2, but the output is high for half of each cycle and low for the other half, generating a square wave output.

In Mode 4 the counter operates as a triggered strobe. After the mode is set the output is high. When the count value is loaded the countdown begins. When a zero count value is reached the output goes low for one timer-clock period. The gate input is normally high, but it may be lowered at any point to inhibit the countdown. The count value must be reloaded for another cycle to begin.

Mode 5 provides a strobe effect triggered by the gate input. The output is normally high. The countdown starts when the gate input goes high, and when the count reaches zero the output drops low for one timer-clock cycle. If the gate input rises, the countdown is restarted.

For the current count value to be read reliably without affecting the operation of a counter, a command must first be issued to the control port requesting that the current residual count value for a specific counter be saved in an internal storage area of the PIT.

The PIT is designed to allow the user to read the current count value at any time, thereby determining how long it will be until a zero count will be reached. For this to be done reliably without affecting the operation of a counter, a command must first be issued to the control port requesting that the current residual count value for a specific counter be saved in an internal storage area of the PIT. The PIT does this without disturbing any ongoing countdown. Two read commands are then issued to the port for the counter to retrieve the two bytes of the saved count. These read commands can be issued at any time after the save operation has been performed. The only restriction is that both read commands must be issued before any other use, such as loading in a new count value, is made of the counter's port.

HOW THE PIT IS USED

As we mentioned earlier, all three counters in the PIT are driven from the CPU clock divided by a factor of four, so the timer-clock period is 0.84 microseconds.

Counter 0 of the PIT is used by BIOS to provide the primary time-of-day service for the PC. It is initialized by the BIOS power-on sequence to run in Mode 3—binary counting—using a full count of 65536. The gate input is wired permanently high. The square-wave output of the counter drives the IRQ0 input of the PC's 8259 Programmable Interrupt Controller (PIC). An interrupt is presented to the CPU so that it can be processed by the BIOS timer interrupt handler every 54.9 milliseconds, or 18.2 times a second.

Counter 1 is used to drive the Direct Memory Access (DMA) refresh function. The storage devices used in the PC do not generally have true static storage capability, and they must be used on a regular basis if the information is not to be lost. To prevent this loss, circuitry on the PC system board forces a dummy read cycle every 15 microseconds. Counter 1 is dedicated to this function. It runs in Mode 2 with a count of 18. All processing of its output is done by the DMA circuitry on the system board.

Counter 2 is used for a number of purposes. It can be used to control the speaker; it is used whenever data is moved to or from the cassette interface; and its inputs and outputs are available through I/O ports for setting and reading under program control for user-defined applications.

BIOS SUPPORT FOR THE TIME-OF-DAY COUNTER

The output from Counter 0 is directly wired to the IRQ0 input of the PIC. The PIC presents the Counter 0 interrupts to the CPU as interrupt 8, which is normally handled by a BIOS routine. A 5-byte data area is reserved (at address 0040:006C) in the PC's main storage for manipulation by this routine. It consists of two words, identified in the BIOS listing as TIMER_LOW and TIMER_HIGH, and a byte labeled TIMER_OFL. TIMER_LOW is incremented by the interrupt handler on every interrupt from Counter 0. TIMER_HIGH is incremented whenever TIMER_LOW reaches its maximum value of 65536 and turns over to zero. Since Counter 0 also runs with a count value of 65536, these two words may be thought of as simple high-order extensions of Counter 0. The timer fields are managed by BIOS and DOS so that the current (4-byte) value represents the elapsed time from the preceding midnight, measured in units of 54.9 milliseconds. This is convenient, since TIMER_HIGH is incremented every 3599.59 seconds, making its value almost exactly equal to the hour of the day. (It would be slightly more convenient for timing if this turned out to be exactly 3600 seconds, which would be the case if

the PC's crystal oscillator were chosen to have a frequency of 14.316558 MHz. The actual frequency used, 14.31818 MHz, seems to have been chosen so that after division by four the color-burst frequency used in color television is obtained.)

The TIMER_OFL byte is used as a flag value for detecting when midnight has been reached. When this occurs, the date needs to be advanced by one day, and the timer fields need to be reset to zero, representing the start of a new day. This flag is set by the timer interrupt handler if, after incrementing TIMER_LOW, the two fields together represent the value 1573040, the number of timer interrupts in a day. At the same time that the flag is set, the timer fields are reset to zero. Advancing the date and resetting the flag are handled as DOS functions the next time DOS calls INT 1Ah for the current time of day.

The timer interrupt handler provides two additional services. It checks the diskette motor timeout and turns off the motor if that timeout has expired; and it issues an INT 1Ch instruction to evoke a user-exit routine. The 1Ch interrupt is initialized to point to a default routine that simply returns to its caller. This default setting is overridden by background programs, such as PRINT.COM, which must provide service at frequent intervals even though other programs are running at the same time.

An additional interrupt routine is provided by BIOS to allow DOS access for reading and setting the timer fields. This routine is called from assembler code by issuing an INT 1Ah instruction, with AH having a value of zero to read the timer fields and a value of one to set the fields. When the fields are read, TIMER_HIGH is returned in CX, TIMER_LOW in DX, and AL contains the flag indicating if midnight has been reached. When a set is done, the fields TIMER_HIGH and TIMER_LOW are set from CX and DX respectively.

It may be of interest to some users that the BIOS equate for the number of timer counts per day is 1573040, whereas the correct value is more nearly 1573042. Because of this, the PC says that midnight has been reached each day about one-tenth of a second earlier than it actually has, which is an error on the order of one part in a million. However, it is not clear that the PC's internal clock should be regarded as providing this level of accuracy in the first place, particularly over extended periods of time and environmental change such as temperature shifts.

The comments in the BIOS listing on the timing loops should also be read with at least a bit of skepticism. There are some places where the CPU is put into a short instruction loop to await some event, usually an interrupt. The comments on the durations of these loops are often wrong, typically by a factor of two.

DOS Support for Time-of-Day Services

The primary DOS support for time-of-day services is provided by function calls 2Ah, 2Bh, 2Ch, and 2Dh. These function calls are described in detail in IBM's Disk Operating System Manual. Function 2Ah returns the current date, which is automatically adjusted to the next day if midnight is reached and makes allowances for the varying numbers of days in each month and for leap years. Function 2Bh sets the current date. Function 2Ch returns the current time in hours, minutes, seconds, and hundredths of a second. Function 2Dh sets the current time. These functions set or reference the BIOS TIMER_HIGH and TIMER_LOW fields for time-of-day purposes.

Under DOS 2.0 there are two other time-related services available. One is function 57h, which allows the date and time fields of a file to be obtained or set. The other is the option of the PROMPT command whereby the day of the week, date, and time of day may be included in the DOS command prompt.

A Simple Timing Test

In examining listing 6, a short COM program that makes a timing test, we may calculate the time that we expect this program to run by looking

The program in listing 1 should take about 22 seconds of elapsed time, based upon the published data for the CPU instruction execution times.

up the expected number of clocks that each instruction should take in one of the standard references on the CPU, multiplying each by the number of times the instruction will be executed and adding the results. This clock total is then multiplied by 0.21 microseconds, the clock period of the CPU in the PC, to obtain the total time. The references state that for typical instruction mixes this calculation should give a value within 5 to 10 percent of the actual time, although much larger variances are possible with certain instruction sequences. (Different references don't always give the same figures for the number of clocks required. Intel's *iAPX86,88* User's Manual is a good source for this information.)

The inner loop of this program requires 29 CPU clock cycles, or 6.09 microseconds, to execute. This times 65536 gives 0.399 seconds for each call on the inner loop; 0.399 times 50 gives 19.96 seconds for the entire program. Add about two seconds to this for DOS command interpretation, storage allocation, program loading,

and other unspeakable overhead items. This implies that this program should take about 22 seconds of elapsed time, based upon the published data for the CPU instruction execution times. This is long enough that an eyeball estimate can easily be made of the actual time the program takes. Such an estimate will likely reveal that the program requires a time closer to 50 seconds, about two and a half times longer than the published times would indicate.

Since the basic complaint is that the existing timer information is too coarse, an alternative is to take the timer interrupts more often.

We should add that we purposely loaded the dice represented by the above program. The ROL instruction is one of several that require less than two clocks per instruction byte. This tends to drain the 4-byte instruction prefetch queue in the CPU, a situation that is known to slow things down. However, measurements of more normal code show similar variances. It is common to observe values as high as 1.5 for the ratio of actual time to calculated time.

When properly measured, such ratios tend to be quite repeatable; the variances are not merely rare random fluctuations. (It is important to be careful about letting extraneous items influence measurements like this. Print spoolers sometimes take over the timer interrupt and run Counter 0 at a rate that is higher than standard, and PRINT.COM uses the timer-interrupt 1Ch user exit for its own special processing. The impact of these actions can be quite large and can invalidate measured data.)

There are three possible sources of delays in the operation of the CPU that might be contributing to these ratios. Note that the CPU is designed so that peripheral devices that cannot keep up with it can request that wait states be introduced, and circuitry for this purpose is included on the PC's system board. One use of this circuitry is by the DMA refresh function driven from Counter 1, which uses about 7 percent of the capacity of the system bus. Another use is to provide wait states requested by the storage units when they are not able to complete read or write actions within the normal four-clock cycle of the CPU. A third use is to provide wait states requested by the "regen" buffers used in the display units.

The above test and many similar ones make it clear that accurate timing information can be obtained only by an experimental approach. Even if it is practical to calculate the expected time, at best this will provide only a lower bound. Ideally, specialized (and expensive) hardware-monitoring devices would be attached to the PC to make the measurements with essentially no interference with its operation. This is clearly not a practical approach for most people, so we must turn to a software approach. As will be seen, in many cases this works entirely satisfactorily.

A First Approach to Improved Timing Information

The information conveniently available to the PC user, the BIOS TIMER_HIGH and TIMER_LOW fields, is perfectly good as far as it goes. As long as there isn't significant interference from other activities in the system to delay the processing of the timer interrupts, the fields will be updated regularly and consistently. The problem is that access to the remainder of the timing information, namely the current residual count in Counter 0, is required at the proper time in order to make a measurement. Because both the timer fields and the counter are 16 bits in length, the current counter value may be regarded as just a low-order extension of the timer fields (negated, however, since the timer counts down, not up).

It is pleasant to find that the designers of the PIT anticipated an interest in obtaining the current count value from the counters. This can be done in a two-step sequence: first, a command is written to the PIT control port to request that the PIT save the desired residual counter value, and second, reads are done from the counter port to retrieve the saved data. It would seem that this is an easy solution to the problem of obtaining higher-resolution data: the current residual count from Counter 0 is simply read and treated as a low-order extension of the BIOS fields.

Life should be so easy. Although the data can easily be read from Counter 0, there is a problem with the retrieved data. The difficulty comes from the way that the PIT generates the square-wave output associated with Mode 3. It first sets the output high and counts down from the starting count by twos, then sets the output low and counts down by twos again. At the end of the second countdown, the IRQ0 interrupt is triggered. The value obtained by reading the counter is therefore ambiguous unless it can be determined which double-speed countdown that value came from. This is determined by whether the counter output is high or low. Unfortunately, the counter output is directly connected only to the IRQ0 input line on the PIC and cannot be interrogated under program control.

Another Approach to Improved Timing Information

Since the basic complaint is that the existing timer information is too coarse, an alternative method might be to take the timer interrupts on a shorter interval. This implies substituting a new timer interrupt handler for the old one, creating a low-order extension to the BIOS timer fields—an extension that is updated on each interrupt—and updating the original BIOS fields when such an update is appropriate. All of the other services of the original interrupt handler must be maintained in addition, so typically the new handler just calls the old one at the original 54.9 millisecond intervals.

Even though this type of action is commonly done by print spoolers, for timing purposes there are problems. One is the placement of the low-order extension of the BIOS fields. If storage within the interrupt handler is used for this purpose, it becomes difficult for a timing routine to access the data because interrupt handlers run as anonymous DOS routines with no natural mechanisms for communication with other routines. Some BIOS storage, such as part of the cassette data area adjacent to the timer fields, could be stolen for this purpose, but such actions are a definite violation of BIOS standards. (Tampering with one of the primary interrupts of the system, of course, is also a violation. Such tampering is not exactly a step to be recommended to the beginning programmer, and it should always be kept in mind that such steps are subject to the vagaries of the next version of DOS, of the system board, of BIOS, etc.)

Another difficulty with running at a higher timer-interrupt rate is the increased overhead. A measurement on the BIOS handler shows that it requires about 135 microseconds to run. At an interrupt interval of 54.9 milliseconds that's an overhead of 0.25 percent. However, suppose the rate is increased by a factor of 64 so that millisecond resolution, which is still fairly coarse, can be obtained. Assuming the new timer interrupt handler takes about as long to run as the old one, the overhead is now 16 percent. This might possibly be an acceptable figure, but further steps in this direction would obviously raise the overhead to an unacceptable amount.

A Successful Approach to Improved Timing Information

The difficulty with the first approach above is that in Mode 3 the counter counts down twice by twos, which makes the information available ambiguous. A direct approach to this problem is to investigate the possibility of running the timer in a mode in which this complication does not appear. Mode 2 seems a likely candidate since it differs from Mode 3 primarily in counting down by ones and in generating an output that is low for only one timer-clock interval of the cycle. This indeed does seem to work very well, perhaps mostly as a result of good luck, of course. For instance, if the output in Mode 2 were inverted and went high (rather than low) for the single timer-clock interval, this would almost certainly not be a wide enough pulse to drive the PIC properly. The mode switch has no effect on the standard services dependent upon the timer.

Assembler is a natural language for this sort of work. We've built three routines: one to set Counter 0 of the PIT running in Mode 2, one to read the current (extended) counter values, and a skeleton control routine to call the set and read routines and the code to be tested. Listing 1 shows the timer-set routine, TIMERS01, which simply writes the necessary information out to the PIT to put Counter 0 in Mode 2. This routine is organized as a NEAR procedure. It is advisable always to call TIMERS01 to set Counter 0 as the first action in a test sequence, in order to insure against the possibility that some other process has reset the timer since it was last set by TIMERS01.

Listing 2 shows the read routine, TIMERR02, which also is organized as a NEAR procedure. It collects the current residual count of Counter 0 and the values in the BIOS timer fields and returns them in registers AX, BX, and CX. The three registers may be thought of as returning a single 6-byte value with a resolution of 0.84 microseconds, as compared to 54.9 milliseconds when using only the BIOS fields. The residual count is obtained by instructing the PIT to save the count in its internal storage and reading the saved value from the counter port. The BIOS timer fields are moved directly from low storage.

It is best to call TIMERS01 to set Counter 0 as the first action in a test sequence, to insure against the possibility that some other process has reset the timer since it was last set by the program TIMERS01.

It is critical that the three values be collected in a way that gives consistent information. Suppose, for instance, that a timer interrupt were to

be processed by the CPU between the two MOV instructions that pick up the TIMER_LOW and TIMER_HIGH values. The value from TIMER_LOW would represent a value from *before* the interrupt processing, and the value from TIMER_HIGH would represent a value from *after* the interrupt processing. While this is not likely to happen often, such a situation might cause large errors in the result. This problem is avoided by blocking interrupts during the critical section of code by surrounding that code with a pair of CLI and STI instructions.

We have found these programs essential for comparing alternative algorithms and for writing time-critical programs; other users should find the techniques equally valuable.

The skeleton control routine, TIMERT03, is shown in Listing 3. This is organized as a FAR procedure to be run as an EXE module. The program is intended to be run under DEBUG with a breakpoint at the location labeled HALT so the timing results in the registers may be examined from the DEBUG display.

An important element of the TIMERT03 program is a standard technique for removing the measurement bias from the result, bias caused by the measurement code itself. This is done essentially by making two runs: one run, which does not contain the code to be tested, to evaluate the time required by the measurement process itself; and another run, which does contain that code, to evaluate the sum of the measurement time and the time required by the code to be tested. The first run is called the calibration pass; the second run is called the test pass. The difference between the results of the two passes yields the number of timer clocks required by the code being tested. (In hexadecimal, unfortunately. The steady user of this code will quickly find that a good hexadecimal calculator, such as the Hewlett-Packard 16C, is an extremely useful tool.)

In practice, this technique seems to give quite good results. Repeated runs tend to match up very well. For instance, if the skeleton program is run without any test code being added, the result usually comes out within a timer tick or two of the expected value of zero. Except perhaps for very short code sequences, such as those in which the time being measured approaches the available resolution, there seems to be no benefit to be gained from putting a loop around the code to be tested and then calculating an average for the result.

It is necessary to watch out for extraneous influences such as background tasks like PRINT.COM and spoolers or other routines that may be fiddling with the timer while a test is being made. If timer interrupts are taken while a routine is being tested, each interrupt will add some 135 microseconds to the time, which is, considering the timer-clock period of 0.84 microseconds, about 160 timer ticks (in hexadecimal, A0h). If the routine being timed involves activity on electromechanical peripherals, such as diskette drives, many random factors will come into play, producing variations in the results.

EXPERIMENTAL RESULTS

In the examples that follow we obtain a calculated time for an instruction sequence by adding up the individual clock values for each instruction in the sequence as taken from the published CPU instruction timings. The measured values are obtained by invoking the code to be tested using the timer-measurement programs described previously.

Listing 4 shows REGCNV04, a typical program tested by this method. It is a simple formatting routine that takes the contents of register AX and converts it to four ASCII hexadecimal digits. The test consists of an execution of this routine preceded by three MOV instructions for setting up the arguments and a CALL instruction to invoke it. The calculated time for this sequence is 74 microseconds; the measured time is 120 microseconds, 62 percent greater.

This seems like a fairly high ratio, and an examination of the code for possible reasons will reveal an ominous sequence of four ROL instructions in the inner loop. As mentioned earlier, these instructions have a tendency to drain the CPU instruction-prefetch queue. A single ROL instruction, with a shift value of four, could be used here, but such an instruction would require 24 clocks. The sequence of four one-bit shift instructions, on the other hand, requires two clocks for each instruction, or eight clocks. In addition, some clocks for setup for the single-shift instruction have to be included.

Listing 5, REGCNV05, shows how the program can be revised to see if the preceding comparison is indeed valid. The results are quite interesting. The calculation for this version gives 95 microseconds, and the measured time is 113 microseconds, only 19 percent greater. In fact, contrary to the analysis above, this approach requires *less* time than the first approach. (Again, these measurements are repeatable; we are not discussing occasional variations.) The performance of the CPU as used in the PC certainly appears to be very sensitive to instruction sequences that drain the prefetch queue.

This approach is not limited to

user-written routines such as the examples we've discussed above. The timer interrupt handler was measured by using an INT 8h instruction as the code to be tested. This isn't quite right, as calling an interrupt handler by the INT instruction requires fewer clocks than a true external interrupt, but it's close.

It is also easily possible to measure times required by DOS function calls. Suppose a user wishes to know if it is safe in a time-critical routine to use function 9 to display a message on the screen. In this case, since DOS code is involved, there is no way to calculate the expected time. However, measuring such a call shows that a short message usually takes approximately 81 milliseconds.

Again, be wary of results that may change drastically if there is a change in the underlying code, such as a new release of DOS. Another source of surprises is that applications may behave differently depending upon the amount of free storage available to them. You might get different results depending on the number of buffers allocated to DOS, the size of the storage available on the PC, or the length of a message string that is passed as a parameter.

SUMMARY

This discussion should help indicate some of the pitfalls to be avoided when extending the use of a system beyond the expressed intent of the original designers. With an idea of how the 8253 timer chip works and how it is utilized in the PC, it is possible to obtain timing information with microsecond, rather than millisecond, resolution. The listings show actual working programs for doing this. We have found the information these programs provide essential for comparing alternative algorithms and for writing time-critical programs; other users should find the techniques equally valuable.

References

Bradley, David J., *Assembly Language Programming for the IBM Personal Computer*, Prentice-Hall, 1984

Disk Operating System, Personal Computer Computer Language Series, version 2.00, IBM Corp., 1983

Eggebrecht, Lewis C., *Interfacing to the IBM Personal Computer*, Howard W. Sams & Co., 1983

iAPX 86,88 User's Manual, Intel Corp., 1981

iAPX 88 Book, Intel Corp., 1981

Peripheral Design Handbook, Intel Corp., 1980

Technical Reference, Personal Computer Hardware Reference Library, revised edition (2.02), IBM Corp., 1983

Willen, David C. and Jeffrey I. Krantz, *8088 Assembler Language Programming: The IBM PC*, Howard W. Sams & Co., 1983

Bob Smith is president of Qualitas, Inc., an IBM PC software vendor located in Bethesda, Maryland. Tom Puckett is a senior systems programmer for STSC, Inc. in southern New Mexico. His primary responsibility is support software for STSC's products for the IBM PC and similar machines.

LISTING 1: TIMERSO1.ASM

```
        title   TIMERS01 -- High Resolution Timer Setup
        page    60,120

        name    TIMERS01        ; module

comment |                   Module Specifications

Copyright: None.

Environment: IBM PC, tested under DOS 2.0.

Segmentation: Program segment CODE, public, byte aligned, class ''.

Public symbols and external references: See Symbols section of listing.

Link requirements: None, standalone subroutine.

Program derived from: None.

Original code by: Bob Smith and Tom Puckett, October 1983.

Modifications by: None.


                        Procedure Specifications

Public Procedure TIMERS01 -- High Resolution Timer Setup

        This is a subroutine to set up Counter 0 of the 8253 to run in Mode 2.
        It normally runs in Mode 3, which makes its residual count values
        useless for high resolution timing purposes since in this mode it counts
        down by twos, with two countdowns per cycle.  In Mode 2 there is a
        single countdown per cycle, by ones, so the residual count values are
        meaningful. (See routine TIMERR02 for recovering timing information.)

        Assumptions: Counter 0 is to be set to normal count of 65536.

        Linkage: Near call and return.

        Arguments: None.

        Effects: Counter 0 set to mode 2, count 65536, binary.

        Results: Flags destroyed.

        Return conditions: None.

        Limitations: None.

|
        page
code    segment public byte
        assume  cs:code

; equates....

timer_0 equ     40h             ; 8253 counter 0 port
timer_ctl equ   43h             ; 8253 control port
timer_set equ   00110100b       ; 8253 Counter 0, set LOB/HOB, mode 2, binary

        public  TIMERS01
TIMERS01 proc   near            ; see specifications at head of listing

        push    ax              ; preserve
        push    cx

        mov     al,timer_set
        out     timer_ctl,al    ; set Counter 0 parameters, prime load trigger
        xor     al,al           ; zero will give full count of 65536
        nop                     ; insure 5 clocks for 8253 recovery time
        out     timer_0,al      ; load low order byte
        nop                     ; 8253 recovery time
        nop                     ;  "       "      "
        out     timer_0,al      ; load high order byte
```

```
        mov     cx,20000        ; must be sure previous countdown has completed,
loop:   loop    loop            ; so spin here 70 milliseconds to guarantee it

        pop     cx              ; restore
        pop     ax
        ret

TIMERS01 endp

code    ends                    ; end code segment

        end                     ; end module
```

LISTING 2: TIMERRO2ASM

```
        title   TIMERR02 -- Read High Resolution Timer Information
        page    60,120

        name    TIMERR02        ; module

comment |               Module Specifications

Copyright: None.

Environment: IBM PC, tested under DOS 2.0.

Segmentation: Program segment CODE, public, byte aligned, class ''.

Public symbols and external references: See Symbols section of listing.

Link requirements: None, standalone subroutine.

Program derived from: None.

Original code by: Bob Smith and Tom Puckett, October 1983.

Modifications by: None.

                        Procedure Specifications

Public Procedure TIMERR02 -- High Resolution Timer Read Routine

        This is a subroutine to return high resolution timing information
        obtained from the BIOS TIMER_LOW and TIMER_HIGH fields, extended
        with low order information obtained from the current residual count
        of Counter 0 of the 8253.  (See routine TIMERS01 for initializing
        Counter 0.)

        Assumptions: Counter 0 running in Mode 2, with full count of 65536.
                     Locations of BIOS TIMER_LOW and TIMER_HIGH fields are
                        40:6C and 40:6E respectively.

        Linkage: Near call and return.

        Arguments: None.

        Effects: None.

        Results: Flags destroyed.
                 AX = TIMER_HIGH.
                 BX = TIMER_LOW.
                 CX = low order extension obtained by reading 8253.

        Return conditions: None.

        Limitations: None.
        page

bios_data_seg   equ     40h
bios_data       segment at 40h
                org     6ch
timer_low       dw      ?
timer_high      dw      ?
bios_data       ends

code    segment public byte
        assume  cs:code

; equates....

timer_0         equ     40h     ; 8253 channel 0 port
timer_ctl       equ     43h     ; 8253 control port
timer_0_latch   equ     00h     ; 8253 command to save channel 0 current count

        public  TIMERR02
TIMERR02 proc   near            ; see specifications at head of listing

        push    ds
        mov     ax,bios_data_seg        ; get pointer to BIOS data area
        mov     ds,ax
        assume  ds:bios_data

        mov     al,timer_0_latch        ; timer 0, latch

    cli                         ; hold off interrupts while fetching data
        out     timer_ctl,al    ; latch current count in 8253
        mov     bx,timer_low    ; get matching values...
        mov     cx,timer_high
        in      al,timer_0      ; read in low order byte
        mov     ah,al           ; tuck it out of the way
        nop                     ; insure 5 clocks for 8253 recovery time
        in      al,timer_0      ; read in high order byte
    sti                         ; allow interrupts again

        xchg    ah,al           ; get in right order in register
        neg     ax              ; get up count from down
        xchg    ax,cx           ; return in common register order
        pop     ds              ; restore
        ret

TIMERR02 endp

code    ends                    ; end code segment

        end                     ; end module
```

LISTING 3: TIMERTO3ASM

```
        title   TIMERT03 -- High Resolution Timer Test Skeleton
        page    60,120

        name    TIMERT03        ; module

comment |               Module Specifications

Copyright: None.

Environment: IBM PC, tested under DOS 2.0.

Segmentation: Program segment CODE, public, byte aligned, class ''.
              Stack segment STACK, stack, paragraph aligned, class ''.
              Data segment DATA, public, byte aligned, class ''.

Public symbols and external references: See Symbols section of listing.

Link requirements: EXE module, must be linked with TIMERS01 and TIMERR02,
                   and any external code to be included in test.

Program derived from: None.

Original code by: Bob Smith and Tom Puckett, October 1983.

Modifications by: None.

                        Procedure Specifications

Program TIMERT03 -- High Resolution Timer Test Program Skeleton

        This is the skeleton of a control routine to be used in running
        high resolution timing tests.  The code to be tested is included
        in this module or called from it.  This routine calls subroutines
        TIMERS01 and TIMERR02 for setting and reading Counter 0 of the 8253.
        It is normally run under DEBUG to allow a breakpoint to be set to
        examine the results of the timing test.

        Assumptions: None.

        Linkage: DOS command.

        Arguments: None.

        Effects: 8253 Counter 0 reset by routine TIMERS01.

        Results: None.
```

```
        Return conditions: None.

        Limitations: Gives incorrect results if test run spans midnight.

        page

stack   segment stack para

        db      32 dup ('stack   ')     ; you may wish to increase this if
                                        ;   the routine being tested makes
stack   ends                            ;   heavy demands on the stack


data    segment public byte

a_cal_high      dw      ?               ; first timer readings...
a_cal_low       dw      ?
a_cal_ext       dw      ?

b_cal_high      dw      ?               ; second timer readings....
b_cal_low       dw      ?
b_cal_ext       dw      ?

;       other storage needed by routine being tested can be included here...

data    ends


code    segment public byte
        assume  cs:code

        extrn   timers01:near   ; 8253 Counter 0 set routine
        extrn   timerr02:near   ; 8253 Counter 0 read routine

        public  TIMERT03
TIMERT03 proc   far             ; see specifications at head of listing

        push    ds              ; make return linkage
        xor     ax,ax
        push    ax
        mov     ax,data         ; get data segment addressible
        mov     ds,ax
        assume  ds:data
        call    tester          ; do the work
        ret

TIMERT03 endp
        page

tester  proc    near

        call    timers01        ; get 8253 counter 0 running in Mode 2

        call    timerr02        ; begin calibration pass

        mov     a_cal_ext,cx
        mov     a_cal_low,bx
        mov     a_cal_high,ax   ; initial extended timer count now saved

calib_begin:
;               put here any test code whose effect is to be removed
;               from the results, such as loop control statements...
calib_end:

        call    timerr02        ; end of calibration pass, begin test pass
        mov     b_cal_ext,cx
        mov     b_cal_low,bx
        mov     b_cal_high,ax

test_begin:
;               put test code here...
test_end:

        call    timerr02        ; end of test pass

        add     cx,a_cal_ext    ; calculate (C-B)-(B-A) as ((C+A)-B)-B...
        adc     bx,a_cal_low    ; ADC to allow for carry in preceding ADD
        adc     ax,a_cal_high   ; now have C+A

        sub     cx,b_cal_ext
        sbb     bx,b_cal_low
        sbb     ax,b_cal_high   ; now have (C+A)-B
        sub     cx,b_cal_ext
        sbb     bx,b_cal_low
        sbb     ax,b_cal_high   ; now have ((C+A)-B)-B in timer counts as
                                ;   a six byte field in AX, BX, and CX

halt:                           ; put DEBUG breakpoint here to examine results
        ret

tester  endp

code    ends

        end     TIMERT03
```

LISTING 4: REGCNVO4

```
        title   REGCNV04 -- Convert contents of AX to hex characters.
        page    60,120

        name    REGCNV04        ; module

comment |               Module Specifications

Copyright: None.

Environment: IBM PC, tested under DOS 2.0.

Segmentation: Program segment CODE, public, byte aligned, class ''.

Public symbols and external references: See Symbols section of listing.

Link requirements: None, standalone subroutine.

Program derived from: None.

Original code by: Bob Smith and Tom Puckett, October 1983.

Modifications by: None.

                        Procedure Specifications

Public Procedure REGCNV04 -- Convert value in AX to ASCII characters.

        This is a subroutine to format and place in storage four ASCII
        characters giving a hexadecimal representation of the value
        passed in AX.

        Assumptions: None.

        Linkage: Near call and return.

        Arguments: AX contains value to be formatted.
                   ES:DI point to destination where result is to be placed.

        Effects: None.

        Results: Flags and AX destroyed.
                 DI incremented by four.
                 Four hexadecimal digits formatted as ASCII characters
                   placed at ES:DI.

        Return conditions: None.

        Limitations: None.

|
        page

code    segment public byte
        assume  cs:code

hextable db     '0123456789ABCDEF'      ; for XLAT below


        public  REGCNV05
REGCNV05        proc    near    ; see specifications at head of listing

        push    bx              ; preserve
        push    cx
        push    dx
```

```
        cld                     ; for STOSB below
        mov     cl,4            ; for ROL below
        mov     dx,ax           ; value to be formatted
        mov     ah,4            ; four hex digits to convert
        mov     bx,offset hextable      ; translate table

digitloop:
        rol     dx,cl           ; the fast way
        mov     al,dl
        and     al,0fh          ; mask off what's not wanted
        xlat    hextable        ; convert AL to ASCII
        stosb                   ; stuff AL at ES:DI, update DI
        dec     ah              ; loop counter
        jnz     digitloop

        pop     dx              ; restore
        pop     cx
        pop     bx
        ret

REGCNV05 endp

code    ends            ; end code segment

         end            ; end module
```

LISTING 5: REGCNVO5

```
        title   REGCNV05 -- Convert contents of AX to hex characters.
        page    60,120

        name    REGCNV05        ; module

comment |               Module Specifications

Copyright: None.

Environment: IBM PC, tested under DOS 2.0.

Segmentation: Program segment CODE, public, byte aligned, class ''.

Public symbols and external references: See Symbols section of listing.

Link requirements: None, standalone subroutine.

Program derived from: REGCNV04.

Original code by: Bob Smith and Tom Puckett, October 1983.

Modifications by: Bob Smith and Tom Puckett, October 1983.
                  Sequence of four ROL DX,1 replaced by single ROL DX,CL.


                        Procedure Specifications

Public Procedure REGCNV05 -- Convert value in AX to ASCII characters.

        This is a subroutine to format and place in storage four ASCII
        characters giving a hexadecimal representation of the value
        passed in AX.

        Assumptions: None.

        Linkage: Near call and return.

        Arguments: AX contains value to be formatted.
                   ES:DI point to destination where result is to be placed.

        Effects: None.

        Results: Flags and AX destroyed.
                 DI incremented by four.

                Four hexadecimal digits formatted as ASCII characters
                   placed at ES:DI.

        Return conditions: None.

        Limitations: None.

        page
code    segment public byte
        assume  cs:code

hextable db     '0123456789ABCDEF'      ; for XLAT below


        public  REGCNV04
REGCNV04        proc    near    ; see specifications at head of listing

        push    bx              ; preserve
        push    dx

        cld                     ; for STOSB below
        mov     dx,ax           ; value to be formatted
        mov     ah,4            ; four hex digits to convert
        mov     bx,offset hextable      ; translate table

digitloop:
        rol     dx,1            ; one bit at a time to minimize number of clocks
        rol     dx,1
        rol     dx,1
        rol     dx,1
        mov     al,dl
        and     al,0fh          ; mask off what's not wanted
        xlat    hextable        ; convert AL to ASCII
        stosb                   ; stuff AL at ES:DI, update DI
        dec     ah              ; loop counter
        jnz     digitloop

        pop     dx              ; restore
        pop     bx
        ret

REGCNV04 endp

code    ends            ; end code segment

        end             ; end module
```

LISTING 6: TIMETEST.ASM

```
code segment public byte
     assume     cs:code
     org        100h            ; room for PSP
timetest proc   near
     xor        cx,cx           ; loop 65536 times
     mov        dx,50           ; times 50 again
loop:
     rol        ax,1            ; 2 clock cycles
     rol        ax,1            ; 2   "     "
     rol        ax,1            ; 2   "     "
     rol        ax,1            ; 2   "     "
     rol        ax,1            ; 2   "     "
     rol        ax,1            ; 2   "     "
     loop       loop            ; 17  "     "
     dec        dx
     jnz        loop
     ret                        ; to INT 20h in PSP
timetest endp
code ends
     end        timetest
```

TECH NOTEBOOK 15

Creating Bounce Bar Menus

Menus that use highlighting to make selecting among options easier

JEFF DUNTEMANN

A bounce bar menu appears as a vertical column of text options with one of them highlighted in reverse video. The user "bounces" the highlighted bar up and down the list of choices using the up/down arrow keys. When the bar highlights the desired choice, the user presses RETURN.

Subroutine 10000 will produce a double-spaced bounce bar menu from text options stored in DATA statements. In the "parameter line" (program line 1020), the user must set an *X,Y* coordinate pair to the menu's upper left corner, indicate the number of text options to display, and indicate the number of the text option to highlight initially; he can then RESTORE to the desired DATA statement and call the subroutine.

The variable OLDBOUNCE allows the calling logic to choose which text option comes up as highlighted. This is useful when the subroutine controls the value of a specific variable: if BACKCOLOR stored the background color value, the user would assign the current value of BACKCOLOR to OLDBOUNCE before calling 10000, and the word describing the current background color would come up highlighted on the screen. The subroutine returns a number from 1 to BNUM (the number of text options) in a variable called BOUNCE. This number is that of the text option that was highlighted when the RETURN key was pressed. The user can also press the Escape key, which sets a variable called ESCAPE to 1. This will indicate to the calling logic that no choice was made.

Only the up arrow, down arrow, RETURN, and Escape keys are legal within the subroutine. If the bounce bar is highlighting the bottom text option and the user presses the down-arrow key, the bounce bar "wraps around" to the top text option.

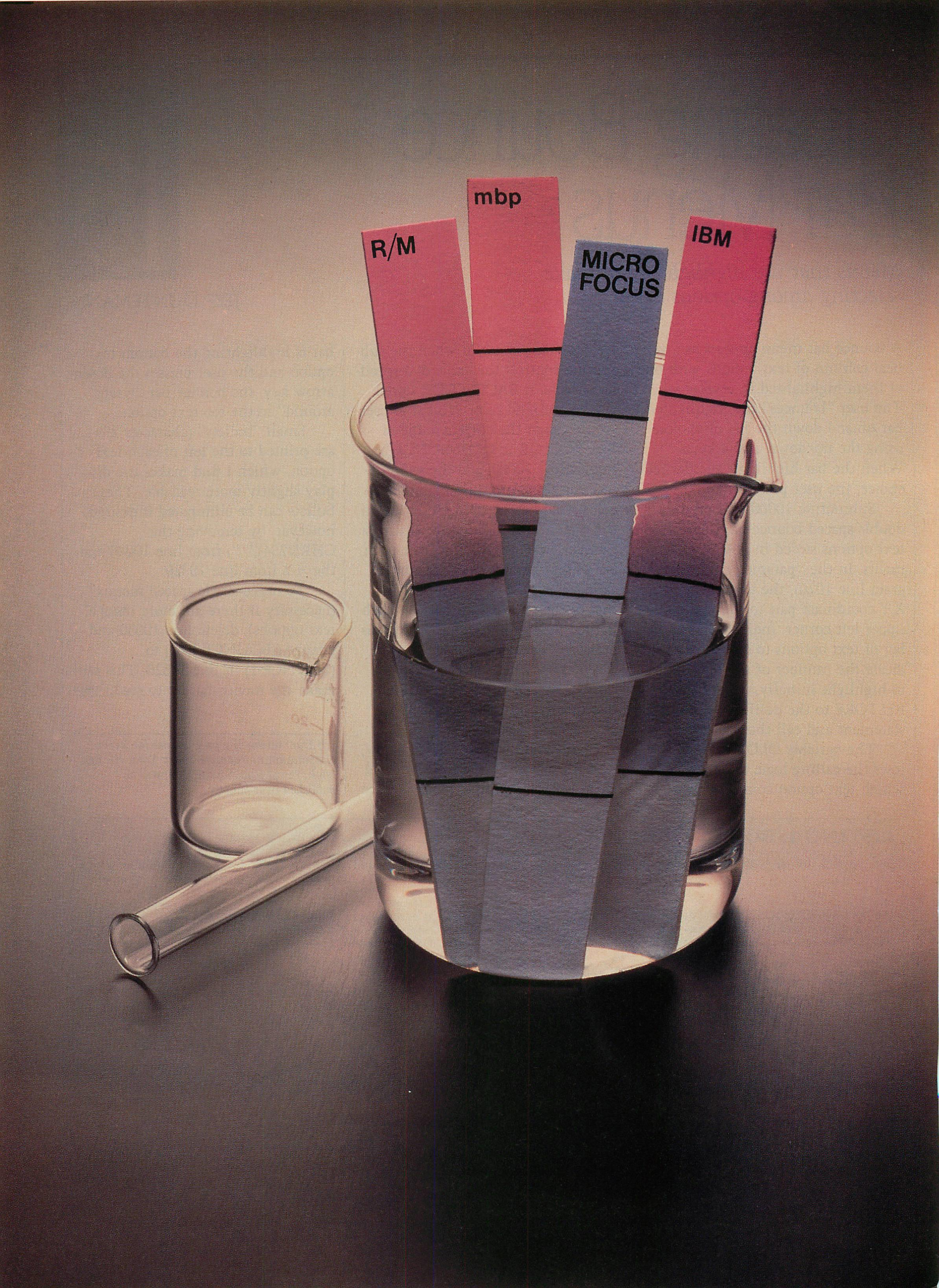
Small "bullets" (character 254) are printed to the left of each text option, which I find makes the display slightly more readable. These bullets can be eliminated if desired, however, by removing the CHR$(254);" "; from line 10040 and the +2 from line 10320.

To make single-spaced menus (necessary if there are more than 11 text options), delete line 10050 and change ((BOUNCE-1)*2) to (BOUNCE-1) in line 10320. This will make the menus harder to read.

Jeff Duntemann is a programmer for Xerox Corporation in Rochester, New York. His first book, Pascal from Square One, *will be published next January.*

LISTING 1: BOUNCE. BAS

```
1000  'Test routine for BOUNCE menu subroutine at 10000
1001  '
1002  '
1010  CLS
1020  BNUM=6:BX=25:BY=7:OLDBOUNCE=3:RESTORE 5050:GOSUB 10000   'Call BOUNCE
1030  IF ESCAPE=1 THEN CLS:PRINT"Aborted Choice":END           'Show choice
1040  CLS:PRINT"Bounce choice=";BOUNCE:END
1998  '
1999  '
5000  'Data statement archive
5001  '
5050  DATA "Set Time and Date","Enter Weekly Timesheets","Issue Paychecks",
"Issue W2 Forms","Update GL Master","Give Boss a Raise"
5999  '
9999  '
10000  '>>>>>Subroutine: Bounce
10010  '
10020  ESCAPE=0:BG=7:FG=0:LOCATE BY,BX,0          'Display menu choices
10030  FOR I=1 TO BNUM
10040    READ BB$(I):PRINT TAB(BX);CHR$(254);" ";BB$(I)
10050    PRINT
10060  NEXT
10070  BOUNCE=OLDBOUNCE:GOSUB 10300                'Highlight old choice #
10079  '
10080  'Keyboard polling loop
10090  '
10100  X$="":WHILE LEN(X$)=0:X$=INKEY$:WEND                 'Get a keypress
10120  IF ASC(X$)=13 THEN COLOR 7,0:RETURN                   'Was it (CR)?
10125  IF ASC(X$)=27 THEN ESCAPE=1:COLOR 7,0:RETURN      'Was it ESC?
10130  IF LEN(X$)<2 THEN SOUND 50,7:GOTO 10100            'Is it Extended?
10170  IF ASC(MID$(X$,2,1))=72 THEN 10240                 'Is it up arrow?
10180  IF ASC(MID$(X$,2,1))<>80 THEN SOUND 50,7:GOTO 10100      'Down arrow?
10189  '
10190  OLDBOUNCE=BOUNCE                                   'Bounce down
10200  IF BOUNCE=BNUM THEN BOUNCE=1 ELSE BOUNCE=BOUNCE+1
10210  SWAP BOUNCE,OLDBOUNCE:GOSUB 10350                  'Unhighlight old
10220  SWAP BOUNCE,OLDBOUNCE:GOSUB 10300                  'Highlight new
10230  GOTO 10100
10239  '
10240  OLDBOUNCE=BOUNCE                                   'Bounce up
10250  IF BOUNCE=1 THEN BOUNCE=BNUM ELSE BOUNCE=BOUNCE-1
10260  SWAP BOUNCE,OLDBOUNCE:GOSUB 10350                  'Unhighlight old
10270  SWAP BOUNCE,OLDBOUNCE:GOSUB 10300                  'Highlight new
10280  GOTO 10100
10290  '
10300  '>>Minisub: Highlight Line
10310  '
10320  COLOR FG,BG:LOCATE BY+((BOUNCE-1)*2),BX+2:PRINT BB$(BOUNCE)
10330  RETURN
10340  '
10350  '>>Minisub: Normalize Line
10360  '
10370  BG=0:FG=7:GOSUB 10300:BG=7:FG=0
10380  RETURN
```


R/M
mbp
MICRO FOCUS
IBM

THE TWELVE FUNCTIONAL MODULES OF ANS COBOL

HOW FOUR COMPILERS COMPLY

CASEY PONTIUS

Micro Focus Level II COBOL, IBM COBOL, Ryan-McFarland Corporation RM/COBOL, and mbp COBOL: a function-by-function guide to how they meet ANS COBOL standards

It is no accident that, 24 years after its inception, COBOL endures as the language of choice for business applications. Its capabilities uniquely meet the data processing needs of business and government, and it is transportable. Because COBOL is a high-level, machine-independent language, there is a COBOL compiler implemented on virtually every business computer.

COBOL was cooperatively designed and standardized by computer users and manufacturers. In 1962 the American National Standards Institute (ANSI) formed a committee whose purpose was to specify the form and interpretation of the COBOL language so that source code could be transported from one computer system to another with a minimum of effort.

Out of this committee's efforts came ANS COBOL X3.23-1968, later revised to ANS COBOL X3.23-1974. ANS COBOL consists of a Nucleus, which contains the elements of the language, and eleven modules, each responsible for performing a specific function: Table Handling, Sequential I/O, Relative I/O, Indexed I/O, Sort-Merge, Report Writer, Segmentation, Library, Debug, Inter-Program Communication, and Communication.

For all modules except the Report Writer two levels are defined. The higher level, Level 2, specifies the implementation of the full set of capabilities for the function. The lower level, Level 1, which is a proper subset of the higher level, defines a basic implementation.

To simplify the matter of determining to what extent a compiler implements the ANS standard, the federal government has further de-

PHOTOGRAPH • NORA SCARLETT

fined a Federal Standard COBOL, which specifies four levels of implementation of the ANS COBOL standard: Low, Low-Intermediate, High-Intermediate and High. In order for a compiler to be used in government contract work, the General Services Administration must certify it as having passed its validation tests and as conforming to one of these four implementation levels.

We reviewed four major compilers for the IBM PC for this Reference Report—IBM COBOL, written by Microsoft; Micro Focus Level II COBOL; Ryan-McFarland Corporation RM/COBOL; and mbp COBOL. Micro Focus Level II is GSA certified at the High Level, which is a full ANS COBOL implementation except for the Report Writer. The other three compilers are GSA certified at the Low-Intermediate Level, which means that they implement Level 1 of the Nucleus, Table Handling, Sequential and Relative I/O, Segmentation, Library, Debug, and Inter-Program Communication modules.

Our investigation of these four compilers focused on how they conformed to the ANS COBOL-1974 specification. In the following Reference Report we discuss our findings by functional module, noting the significant features that each compiler supports as well as any extensions it provides. Where tables are used to summarize features we use the following notations and abbreviations:

IBM—IBM COBOL Version 1.0 written by Microsoft
Micro Focus—Micro Focus Level II COBOL Version 2.1
R/M—Ryan-McFarland Corporation RM/COBOL Version 1.5
mbp—mbp COBOL Version 6.12
yes—supported feature
no—unsupported feature
cmt—language element treated as a comment, no action taken
O—see note n

A "Y" or "N" in the ANS 1 or ANS 2 column indicates whether the feature is required for the compiler to meet the level's standard.

1

Nucleus

The Nucleus contains the language elements that are necessary for internal processing, such as qualifications, punctuation, data-names, connectives, figurative constants, and verbs.

NUCLEUS EXTENSIONS

Data Division. All four compilers contain an extension for data in packed-decimal format (COMPUTATIONAL-3). All bytes, except the rightmost byte in a packed-decimal item, contain two digits with the value 0-9. The rightmost byte in a packed-decimal item contains the item's low-order digit and a sign. For example, the five-byte ASCII value 12345 would become the three-byte value 12345F (hex) in packed-decimal format.

IBM COBOL
COMPUTATIONAL-0—16-bit binary data item

Micro Focus COBOL
Literals may be expressed in hex (X"nn").
FILLER group items are allowed.
Redefinitions of data-names need not be the same length. The compiler will reserve the larger length.

1

Feature	ANS 1	ANS 2	IBM	Micro Focus	R/M	mbp
level numbers 01-49	N	Y	yes	yes	yes	yes
level 66	N	Y	no	yes	yes	yes
level 88	N	Y	[1]	yes	yes	yes
operators +, -, *, /, **	N	Y	yes	yes	[2]	yes
logical AND, OR, NOT	N	Y	yes	yes	yes	yes
use of =,<,>	N	Y	yes	yes	yes	yes
abbreviated combined relational conditions	N	Y	yes	yes	no	yes
sign test	N	Y	yes	yes	no	yes
nested IFs	N	Y	yes	yes	yes	yes
COMPUTE	N	Y	yes	yes	[2]	yes
PERFORM VARYING . . . UNTIL	N	Y	yes	yes	yes	yes
STRING, UNSTRING	N	Y	yes	yes	no	yes
procedure names consisting of digits only	N	Y	yes	yes	yes	yes
qualification of names	N	Y	[3]	yes	yes	yes
CORRESPONDING	N	Y	no	yes	yes	yes
series of procedure names in the ALTER statement	N	Y	no	yes	yes	yes
REMAINDER in DIVIDE	N	Y	no	yes	yes	yes
ENTER (non-COBOL code)	N	N	no	no	no	yes
groups of characters in INSPECT	N	Y	no	yes	yes	yes
multiple destinations in ADD, SUBTRACT, MULTIPLY, DIVIDE	N	Y	no	yes	no	yes
DATE-COMPILED	N	Y	cmt	yes	no	yes
GO TO without procedure name	N	Y	yes	yes	no	yes
COMPUTATIONAL-3	N	N	yes	yes	yes	yes
run time switches	Y	Y	yes	yes	yes	no

[1] yes, with the restriction that either a value list or a range may be specified, but the two may not be intermixed
[2] yes, except exponentiation
[3] yes, in procedure division only

Level numbers need not be in ascending sequence.
The function names SYSIN and SYSOUT may be used. These names can be changed if currently existing application programs will conflict.
RM/COBOL
COMPUTATIONAL-1 — 16-bit binary data item
COMPUTATIONAL-6 — same as COMPUTATIONAL-3, but without sign
mbp COBOL
COMPUTATIONAL-1 — 16-bit binary data item

Source Coding Formats. All four compilers adhere rather strictly to the ANS specified source coding formats, with a few exceptions:
IBM COBOL allows tabs and lowercase letters.
Micro Focus COBOL allows lower case. Sentences may begin in either area A or B.
RM/COBOL and mbp COBOL allow subordinate level numbers in Area A.

Video Extensions. All four compilers provide extensions for interactive screen I/O.
IBM COBOL
A Screen Section has been added to the Data Division for the definition of screen formats. An entire form can be sent to the screen with a single DISPLAY screen-name command. Subsequent ACCEPT screen-name commands receive only the variable data entered by the operator.

Variable fields can be displayed with default data, currently existing file-record data, or simply "prompt" data (i.e., dots for alphanumeric fields, zeroes for numeric fields, decimal point indicators). Foreground/background color designation can be selected. Highlight, reverse-video, underline, blink, and secure (no echo) can be specified for individual fields. An AUTO option provides the ability to skip to the next field when a field has been filled.

Input fields can be directed back to a file-record buffer or working-storage for further processing. Input data is edited to ensure that it matches its picture. Unacceptable data is immediately refused. All action is specified in the Screen Format Section and takes place automatically when the ACCEPT or DISPLAY command is issued.

The entire screen format can be displayed or accepted with a DISPLAY/ACCEPT screen-name command, or any portion of a screen format can be accessed with a DISPLAY/ACCEPT coordinates data-name command.
Micro Focus COBOL
This compiler also uses the ACCEPT/DISPLAY statements. The command formats are

DISPLAY data-name-1 AT data-name-2 UPON CRT

ACCEPT data-name-1 AT data-name-2 FROM CRT

Data-name-1 is a record, group, elementary item, or literal. Data-name-2 is a line/position coordinate.

Data can be underlined if hardware permits. Input data is edited to ensure that it matches its picture. Unacceptable data is refused.

Upon execution of the ACCEPT statement, the cursor is displayed in the first nonfiller field on the screen (unless otherwise specified). When a field is filled, the cursor skips to the next nonfiller field. The operator ends input by hitting carriage return.

A separate product called FORMS-2 is available. FORMS-2 allows the user to design and maintain screen layouts on the CRT. In addition to maintaining the screen layout file, FORMS-2 can generate the source code for the data description required to display the form on the screen. This source code may subsequently be accessed by a COBOL program using the COPY statement.
RM/COBOL
RM/COBOL uses the ACCEPT and DISPLAY statements with multiple operand fields to achieve interactive screen displays.

Phrases used with the ACCEPT/DISPLAY statements control cursor positioning, maximum input length, input data echoing, and optional display of prompt text and current field contents. Reverse-video, high or low-intensity, and blink can be specified.

Input data is edited to ensure that it matches its picture; the ON EXCEPTION phrase is used for error-detection handling.
mbp COBOL
mbp COBOL's approach to interaction video I/O is a Screen Management System, which has two parts:
1. A stand-alone Mask Editor program enables the programmer to generate and update fixed-screen formats (called screen masks) in a Mask Library. Field position, size, type, usage, control, and video attributes are assigned via the Mask Editor.
2. A Screen Handler in the COBOL library enables the application program to send and receive variable data via subroutine calls:

SCRNOPEN	opens the Mask Library file.
SCRNINIT	finds the specified screen format in the Mask Library, displays it, and initializes the variable-data portion of the screen and the data buffer in memory.
SCRNWTRD	displays variable data on the screen and accepts updating. Input data is edited to ensure that it matches its picture. Unacceptable data is immediately refused.
SCRNWRIT	displays variable data on the screen.
SCRNPRNT	prints a hard copy of the screen display.
SCRNCLOS	closes the Mask Library file.

Casey Pontius is a software specialist who works in the Baltimore area. She has been programming for twenty years.

2

Feature	ANS 1	ANS 2	IBM	Micro Focus	R/M	mbp
# of dimensions	3	3	3	49	3	3
variable length (OCCURS DEPENDING ON)	N	Y	no	yes	yes	yes
ascending/descending key	N	Y	yes	yes	no	yes
SEARCH, SEARCH ALL	N	Y	yes	yes	no	yes

3,4,5

Feature	ANS 1	ANS 2	IBM	Micro Focus	R/M	mbp
RERUN	Y	Y	cmt	cmt	no	yes
multiple operands in OPEN, CLOSE	N	Y	yes	yes	yes	yes
VALUE OF file-id IS data-name	N	Y	yes	cmt	no	cmt
EXTEND mode for OPEN	N	Y	yes	yes	yes	yes
LINAGE clause	N	Y	yes	yes	no	no
dynamic access mode	N	Y	yes	yes	yes	yes
START WITH KEY EQUAL, GREATER, NOT LESS	N	Y	yes	yes	yes	yes
SAME RECORD AREA	N	Y	yes	yes	yes	yes
OPTIONAL input files	N	Y	no	yes	no	no
ALTERNATE RECORD KEY, DUPLICATES	N	Y	no	yes	yes	yes
RESERVE multiple I/O buffers	N	Y	cmt	yes	no	cmt
CODE-SET	N	Y	cmt	cmt	yes	yes
ORGANIZATION IS LINE SEQUENTIAL	N	N	yes	yes	no	no

10

Feature	ANS 1	ANS 2	IBM	Micro Focus	R/M	mbp
WITH DEBUGGING MODE	Y	Y	yes	yes	no	yes
object time switch	Y	Y	no	yes	no	no
USE FOR DEBUGGING	Y	Y	no	yes	no	yes
DEBUG-ITEM	Y	Y	no	yes	no	yes
debugging lines	Y	Y	yes	yes	[1]	yes

[1] *Debugging lines are supported, but are controlled by a compile time switch rather than the WITH DEBUGGING MODE clause.*

Compilers mentioned in this article:

R/M COBOL
Ryan-McFarland Corp., 609 Deep Valley Dr., Rolling Hills Estates, CA 90274, 213-541-4828
CIRCLE 421 ON READER SERVICE CARD

IBM COBOL
Microsoft Corp., 10700 Northup Way, Bellevue, WA, 98004, 206-828-8080
CIRCLE 422 ON READER SERVICE CARD

MICRO FOCUS COBOL
Micro Focus, Inc., 1601 Civic Center Dr., Santa Clara, CA, 95050, 408-496-0176
CIRCLE 423 ON READER SERVICE CARD

mbp COBOL
Software & Systems Technology, Inc., 7700 Edgewater Dr., Suite 360, Oakland, CA, 94621, 415-632-8668
CIRCLE 424 ON READER SERVICE CARD

2

Table Handling

Table Handling provides the programmer the capability of defining and accessing tables of data items.

ANS Level 1 has the capability of handling three-dimensional fixed-length tables.

ANS Level 2 specifies that the compiler can handle three-dimensional variable-length tables with ascending/descending keys, and permits searching a table dimension for items satisfying a certain condition.

TABLE HANDLING EXTENSIONS

Level II COBOL allows up to 49 dimensions.

3,4,5

Sequential, Relative, and Indexed I/O

ANS COBOL Level 1 of these modules means basically the barebones I/O handler. Level 2 has the goodies, such as EXTEND, DYNAMIC, START, ALTERNATE KEY, OPTIONAL, and so forth.

I/O EXTENSIONS

LINE SEQUENTIAL files contain variable-length records terminated by a carriage return ("printer image"). These files are useful for generating reports on disk to be printed elsewhere.

Micro Focus COBOL

LINE SEQUENTIAL files are supported.

LEVEL II COBOL

LINE SEQUENTIAL files and run-time allocation of file names are supported.

All FD clauses are optional.

Tabbing is available with the WRITE statement.

6

Sort/Merge

The ANS Sort/Merge module allows the inclusion of one or more sorts and/or merges in a program.

Micro Focus is the only compiler that supports this module. Full Level 2 support is provided.

7

Report Writer

The ANS Report Writer provides for the semi-automatic production of printed reports.

It is not implemented in any of the reviewed compilers.

8

Segmentation

The Segmentation Module handles program overlay segments.

ANS Level 1 provides the facility for specifying permanent (always resident) and independent (overlay) segments. All sections with the same segment number and all permanent segments must be contiguous in the source program.

ANS Level 2 provides the facility for intermixing sections with different segment numbers and allows the fixed portions of the program to contain segments that may be overlaid.

IBM COBOL and RM/COBOL provide Level 1 support.

Micro Focus COBOL supports Level 2.

The availability of mbp COBOL's segmentation support is dependent on the DOS Linker's support of overlays. Coded programs using segmentation will compile and run correctly but will not actually be segmented. Compilers distributed prior to DOS Linker overlay support will be updated at no charge.

9

Library

The Library module provides the facility for copying text from a library into a program at compile time.

ANS Level 1 allows text to be copied from a single library without any changes.

ANS Level 2 allows text to be replaced when copying (macro style), and allows the use of more than one library at compile time.

mbp COBOL, RM/COBOL, and IBM COBOL support Level 1.

Micro Focus COBOL supports Level 2.

LIBRARY EXTENSIONS

Micro Focus COBOL allows an external file-name literal to be present in the COPY statement.

10

Debug

ANS COBOL Debug has five features:

1. The compile time switch "WITH DEBUGGING MODE"
2. An object time switch
3. The USE FOR DEBUGGING statement
4. The special register DEBUG-ITEM
5. Debugging lines (identified with a "D" in column 7). Debugging lines contain code that is compiled if the WITH DEBUGGING MODE clause is present or, if the clause is not present, treated as comments.

DEBUG EXTENSIONS

IBM COBOL

IBM COBOL has two extended features for debugging:

1. An EXHIBIT statement that displays data names and values at designated points. For example, the statement "EXHIBIT PART-NUM-INDEX." would display the text "PART-NUM-INDEX 00023."
2. A TRACE statement that displays program procedure names as they are called

Micro Focus COBOL

Micro Focus COBOL markets a separate product called Animator, which has an array of debugger commands and displays the source code on the screen at run time.

RM/COBOL

RM/COBOL has an interactive debugger with breakpoint, step, and dump commands.

mbp COBOL

mbp COBOL has an interactive debugger with trace, step, go, breakpoint, and dump/modify commands.

11

Inter-Program Communication

The Inter-Program Communication module provides the ability to transfer control between programs at run time and allows all programs to have access to common data items.

ANS Level 1 allows both transfer of control to programs whose names are known at compile time and sharing of data between programs.

ANS Level 2 provides the ability to transfer control to programs whose names are not known at compile time, determine the current availability of memory for a program, and release unneeded program space.

IBM COBOL and mbp COBOL support Level 1.

RM/COBOL supports Level 1 and also allows a program whose name is not known at compile time to be called.

Micro Focus COBOL supports Level 2.

INTER-PROGRAM COMMUNICATION EXTENSIONS

IBM COBOL

IBM COBOL provides an extension that allows program chaining. The CHAIN statement is similar in format to the CALL statement, but the latter is generally used by a main program to transfer control to a subprogram; control is expected to be returned. The CHAIN statement is used by a main program to transfer control to another main program, and control is not to be returned. Both CALL and CHAIN may pass parameters to the called program.

Micro Focus COBOL

Micro Focus COBOL will tolerate to some extent a mismatched length CALL/USING operand list. If the CALLing list is longer, the compiler will ignore trailing operands. If the USING list is longer, however, it is important that the trailing operands must not be referenced at run time by the called program. If they are, a run-time error will result.

CALL/USING parameters are not restricted to 01 and 77.

RM/COBOL

Literals are allowed in the USING portion of the CALL statement. CALL/USING parameters are not restricted to 01 and 77.

12

Communication

The ANS Communication module provides for communications through a Message Control System with local and remote devices.

IBM COBOL, RM/COBOL, and mbp COBOL do not support the Communication Module.

Micro Focus COBOL accepts the full Level-2 syntax for the module and produces intermediate code. Run-time support is not currently available for this function.

What should you look for in personal

Before you go looking for personal computer software, you should know what personal computer software looks like.

(You're not alone if you don't already know that software programs come in a package.)

Programs are "pre-recorded" on cartridges, tapes or diskettes. And, although you can't tell by looking at these cartridges, tapes or diskettes, the programs on them can be very different.

What you put in is what you get out.

What happens when you play a high-quality tape on a high-quality recorder? Superior sound.

This analogy can also hold true with software. The better the program quality, the better the result —be it improved productivity or creativity.

IBM Personal Computer Software is both tested and approved by IBM. And these programs are designed to take advantage of an IBM personal computer's many advanced hardware features.

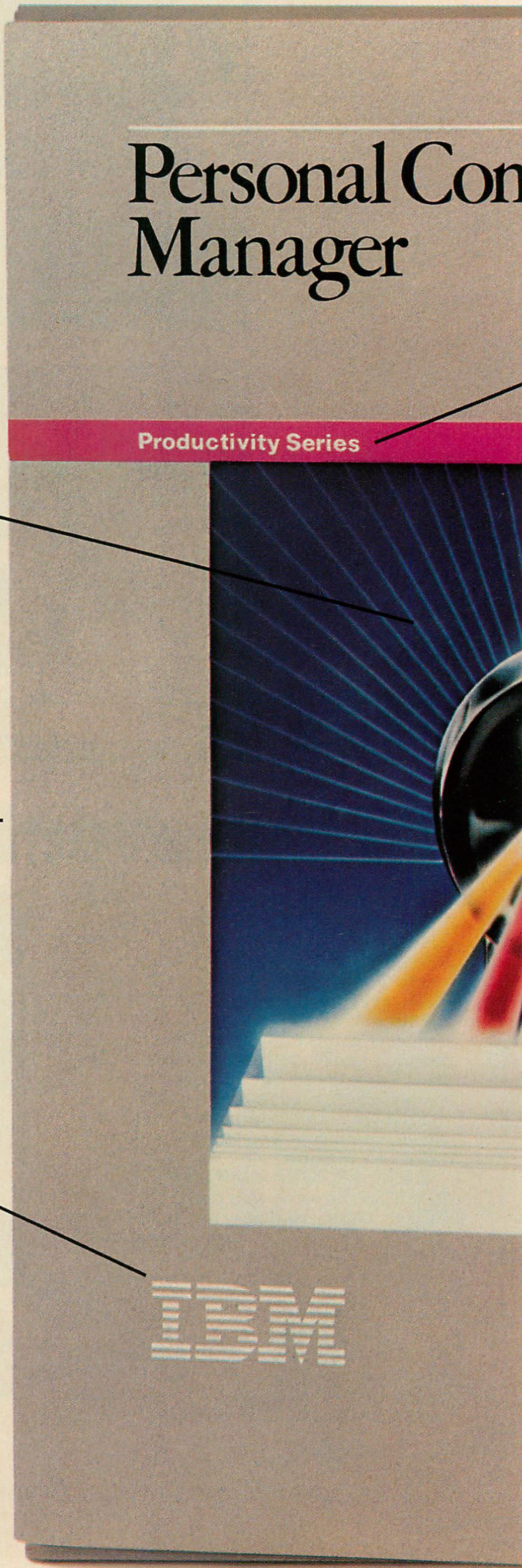
What the value is.

What are improved math skills worth? More efficient inventory control? Faster communications? What is accomplishment worth?

Any way you look at it, the value of personal computer software is the value of doing your best.

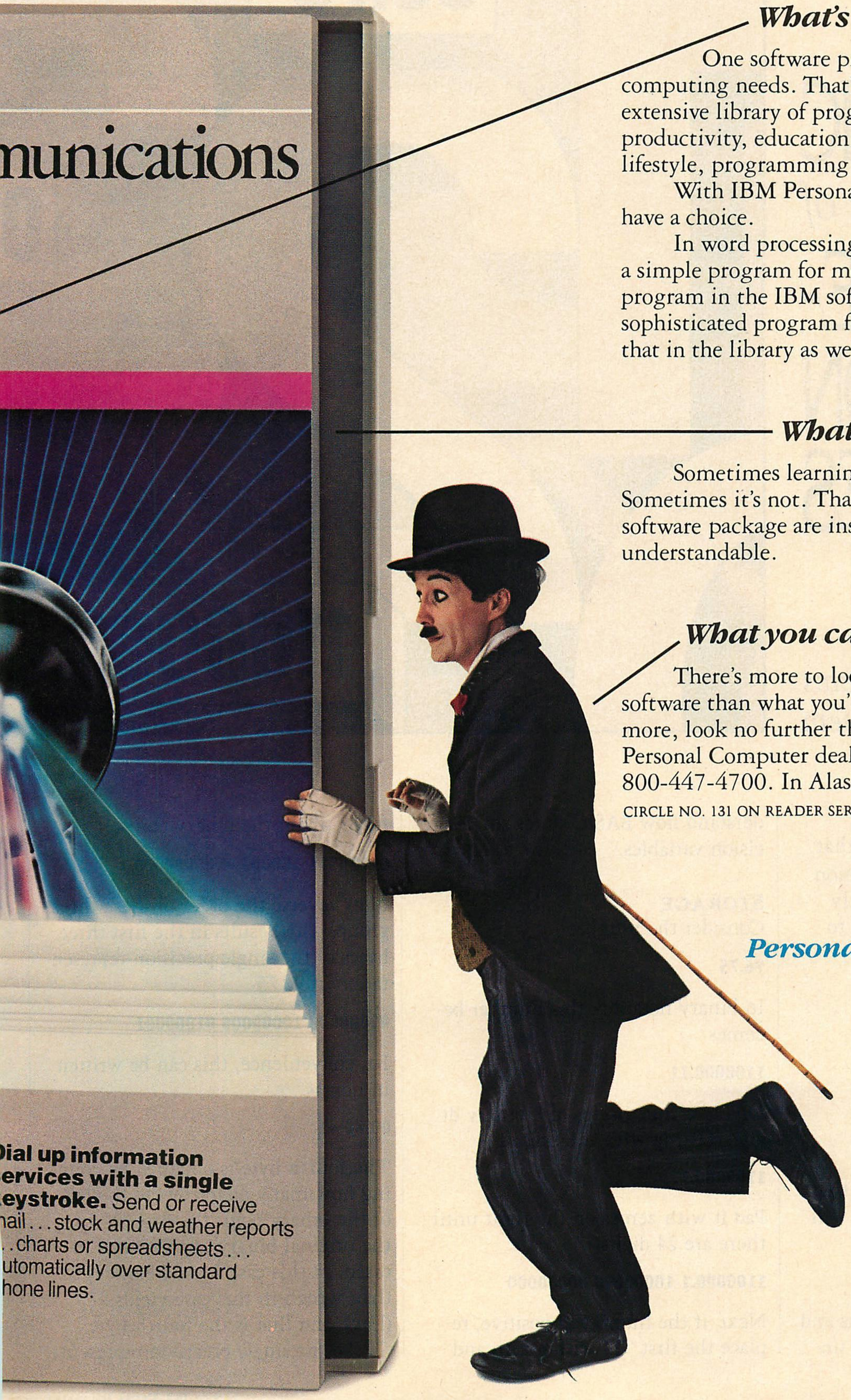
What compatibility means.

Many of the same software programs that run on the IBM Personal Computer you use at work will run on other IBM personal computers you use in other places. So you can, for example, continue in your family room what you started in the classroom or boardroom. (Or the other way around.)



The Little Tramp character licensed by Bubbles, Inc., S.A.

computer software?



What's available.

One software program can't satisfy everyone's computing needs. That's why IBM has such an extensive library of programs. You'll find a series for productivity, education, business, entertainment, lifestyle, programming or communications.

With IBM Personal Computer Software, you have a choice.

In word processing, for example, you may want a simple program for memo writing. You'll find that program in the IBM software library. If you want a sophisticated program for report writing, you'll find that in the library as well.

What's inside.

Sometimes learning a program is easy. Sometimes it's not. That's why inside every IBM software package are instructions that are clear and understandable.

What you can do right now.

There's more to look for in personal computer software than what you've read here. To find out more, look no further than your authorized IBM Personal Computer dealer. For one near you, call 800-447-4700. In Alaska or Hawaii, 800-447-0890.

CIRCLE NO. 131 ON READER SERVICE CARD

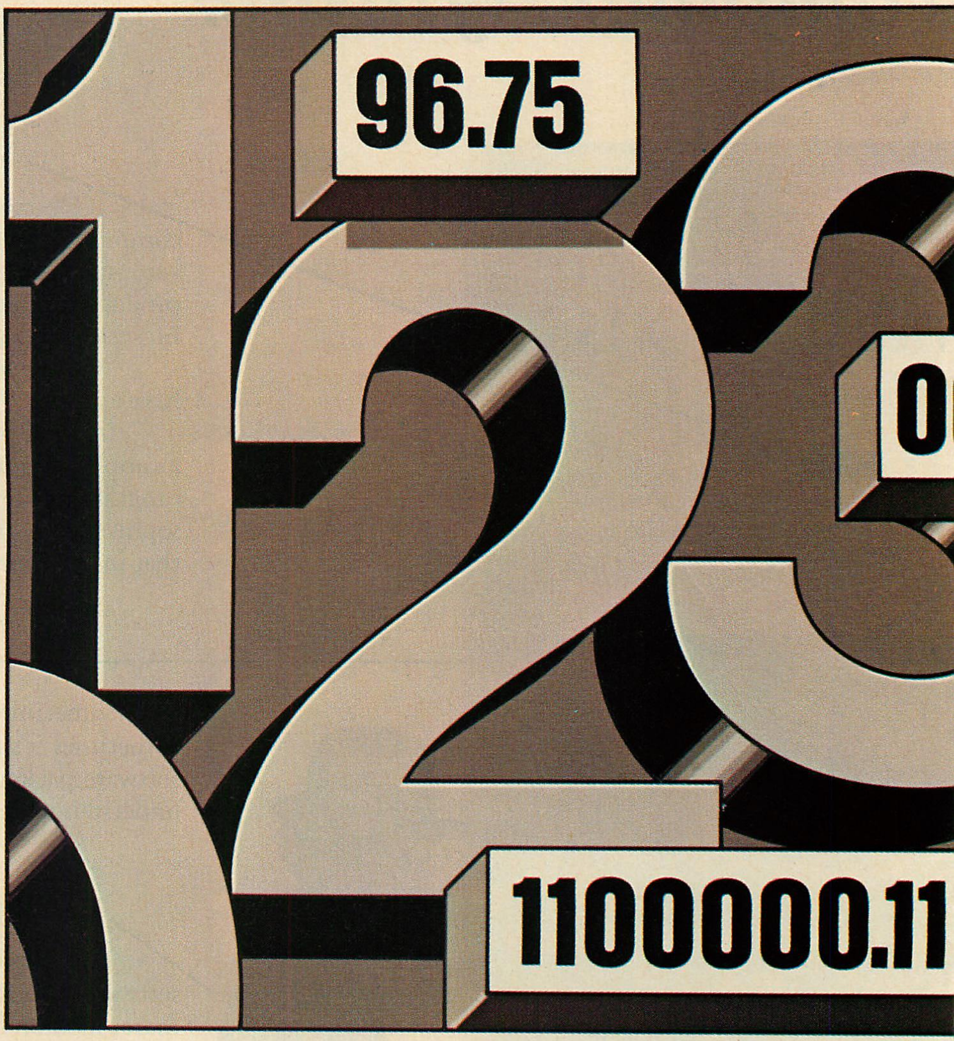
IBM®

Personal Computer Software

MICHAEL MATHER

IBM BASIC *and the storage of* SINGLE-PRECISION VARIABLES

Some common problems and how to solve them



One annoying problem with programming in BASIC is that computations using single-precision variables sometimes yield slightly inaccurate results. For example, to the statement

PRINT 9.893

BASIC will respond

9.893001

and if the statement

10 x = 9.893

is included in a program, when that program is listed, the statement will read

10 x = 9.893001

To understand why this happens and how it can be avoided, we must understand how BASIC stores single-precision variables.

STORAGE

Consider the number

96.75

In binary notation, this number becomes

1100000.11

Divide it into groups of 8 binary digits, starting with the first "1":

1100000.11

Pad it with zeroes on the right until there are 24 digits:

1100000.1 10000000 00000000

Next, if the number is positive, replace the first "1" with a zero and forget about the binary point:

01000001 10000000 00000000

Now reverse the order of these three groups; this results in the first three bytes of the single-precision representation of 96.75:

00000000 10000000 01000001

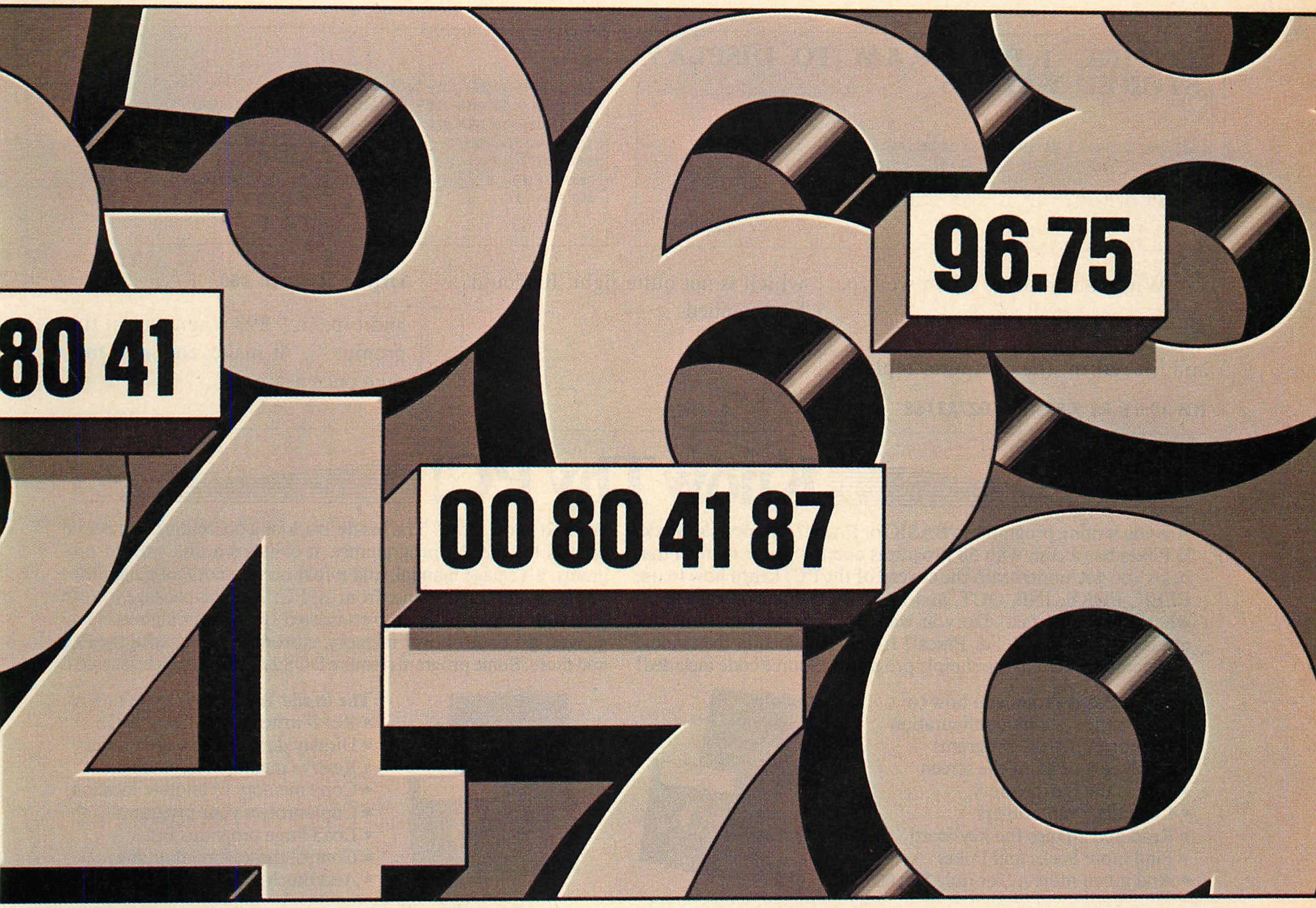
For convenience, this can be written in hex as

00 80 41

The fourth byte is obtained by counting how many characters to the left of the binary point the first "1" in the original binary expression is located, in this case 7. Add 80(hex) to that, which in this case yields 87 (hex), and that is the fourth byte.

So the single-precision representa-

ILLUSTRATION • GERARD HUERTA



tion of *96.75* is

00 80 41 87

APPROXIMATIONS

The first step above involved writing the number in binary form. Unfortunately, for a short decimal number, this may involve continuing forever. For example, 9.893 becomes

1001.1110010010011011101001011l...

In this case we should round after 24 significant digits. If the 25th significant digit is "1" we round up; otherwise, we round down. Breaking the above example after 24 significant digits will result in

1001.11100100100110111010 010111...

and rounding gives

1001.11100100100110111010

which, when we apply the process above, gives the single-precision representation:

BA 49 1E 84

PRINTING SINGLE-PRECISION NUMBERS

When BASIC is asked to print a single-precision number, it converts it into a string of digits with a decimal point and rounds the answer to the seven significant digits.

For example, the number calculated above, which is precisely

1001.11100100100110111010

in binary, is converted by BASIC to 9.892999649047 . . . and rounded to 9.893000. BASIC then drops the trailing zeroes and gets the answer:

9.893

which is back where we started.

WHAT DOES BASIC REALLY DO WITH THESE NUMBERS?

What I have described above is theoretical. In order to see what actually happens in the machine, we will have to do some digging.

First, we will write a routine to display the actual stored value (expressed in hex) and then a decimal approximation to more than seven digits (see listing 1).

Dr. Michael Mather, who lives in Toronto, Canada, is a consultant to the group area of life insurance companies. He is currently helping to design a pension system.

LISTING 1 PROGRAM TO DISPLAY STORED VALUES

```
300 '*** DISPLAY X AS STORED AND IN DECIMAL
310 ADDR% = VARPTR(X)
320 FOR I = 0 TO 3
330 J = PEEK(ADDR% + I)
340 J1 = INT(J/16) :J2 = J MOD 16
350 IF J1>= 10 THEN J1$ = CHR$(J1+55) ELSE J1$ = CHR$(J1+48)
360 IF J2>= 10 THEN J2$ = CHR$(J2+55) ELSE J2$ = CHR$(J2+48)
370 PRINT J1$; J2$; " ";
380 NEXT
390 X# = X
400 PRINT X#
```

With this routine, when we type

x = 9.983 :GOTO 300

and hit return, the computer replies

BB 49 1E 84 9.893000602722168

which is not quite right. It should have replied

BA 49 1E 84

If we try

input x :goto 300

and type in 9.893 when we get the prompt "?," it makes the same mistake once again.

In order to make the machine calculate correctly, it is necessary to force the use of double precision in setting the single-precision variable x. We can do this in several ways:

x = 9.893# :goto 300

or

x = val ("9.893") :goto 300

or

input x# :x=x# :goto 300.

Obviously, BASIC sometimes calculates single-precision variables incorrectly; to avoid this problem, it is helpful to use double-precision variables, if they are used carefully.

DISPLAYING SINGLE-PRECISION NUMBERS

After we have stored the value of x in the example above, we may want to display it. If we say

x = 9.893#

we know that x is stored correctly. But if we print x we get

PRINT x
9.892999

which again is wrong.

Apparently, using PRINT with single-precision variables can also cause problems. Getting around these problems is complicated and depends on the particular circumstance.

One way is to use PRINT USING, which will display in a fixed format. If we type

PRINT USING "#.###"; x

it will print correctly. However, this will print trailing zeros, which may not be desirable. Also, using STR$ with single-precision variables causes the same error to occur and this technique cannot be used to solve the problem in that case.

To avoid the problem when using STR$, a user-defined function may be defined as follows, using FNS$ instead of STR$:

```
DEF FNS$(x) = STR$ (VAL
   (LEFT$(STR$(x*1.0000005#),8)))
```

and use FNS$ in place of STR$. This works for numbers between 1 and 1 million and gives one less decimal place of accuracy.

HOW FREQUENTLY DO THESE PROBLEMS OCCUR?

To find out how frequently BASIC miscalculates single-precision variables when the PRINT statement is used, try running this program:

```
100 FOR i% = 800 to 1000
200   X = i%/100#
300   PRINT
400   NEXT
```

Now try the following program to see how often BASIC makes this mistake when the STR$ statement is used. Enter exactly the same number twice, starting with 9.000, 9.001, 9.002, etc.; the program will beep when the two are stored differently.

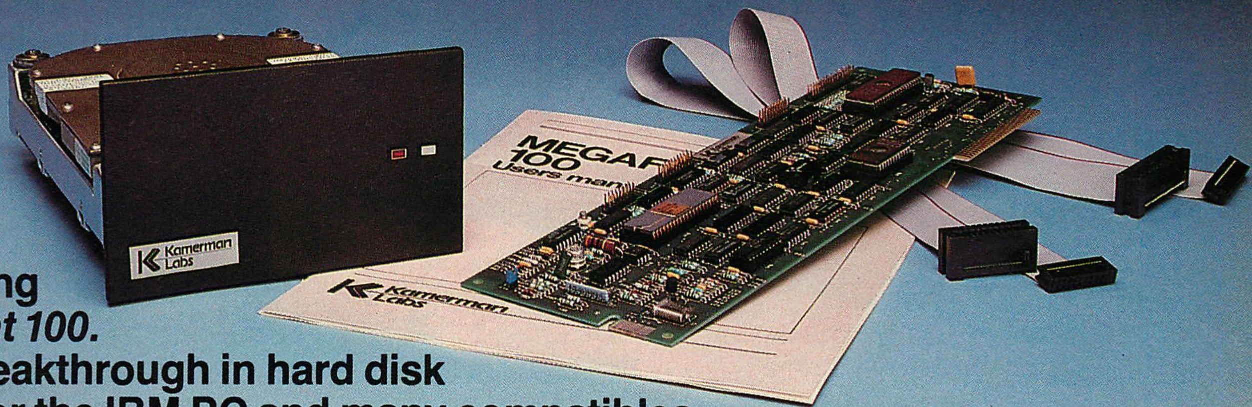
```
10 INPUT x
20 INPUT X$ :X2 = VAL (X$)
30 IF X <> X2 THEN BEEP:
   PRINT X; X2
40 GOTO 10
```

As these programs indicate, problems of this type occur frequently enough that is worthwhile taking some extra time when programming to avoid inaccuracy. The techniques mentioned above should ensure that BASIC programs using single-precision variables will yield accurate results.

FOR THE IBM PC*, COMPAQ*, ETC.

10 Mbyte Hard Disk System

$895



Introducing *Megaflight 100.* A cost breakthrough in hard disk storage for the IBM PC and many compatibles.

At last! A real cost revolution in hard disk storage. The Megaflight 100 is a high-performance, half-height hard disk system which includes a state-of-the-art drive unit, disk controller card and software. You get 10 Megabytes (formatted) of high-speed data storage for less than half the cost of comparable PC hard disk systems!

Full hardware and software compatibility.
Megaflight 100 features full plug compatibility with the IBM XT, requires no extra power and uses IBM's DOS 2.0 or 2.1 software drivers. You're assured of complete compatibility with all future XT releases, and network systems such as PC Net,** Etherseries** and Omninet.**

Installation in less than 20 minutes.
Megaflight 100 includes everything you'll need for internal installation in the PC, Compaq, and others. This includes cables, easy installation instructions and a comprehensive manual. Installation typically takes less than 20 minutes and requires no special technical knowledge.

One year warranty.
Megaflight 100 is covered by a one year warranty on parts and labor.

To order, send check or money order for $895 (includes shipping and handling) to the address listed below. Or use Visa or MasterCard and call our toll free number to order today (add 3% service charge for Visa and MasterCard).
Call toll free **800-522-2237.**
In Oregon, Alaska and Hawaii call 503-626-6877.

Dealer, OEM inquiries invited.
For more information call 503-626-6877.

*IBM is a registered trademark of International Business Machines. COMPAQ is a trademark of COMPAQ Computer Corporation.

**PC Net is a trademark of Orchid Technologies. Omninet is a trademark of Corvus Systems. Etherseries is a trademark of 3Com.

CIRCLE NO. 172 ON READER SERVICE CARD

Kamerman Labs

7787 S.W. Cirrus Dr., Bldg. 26, Beaverton, OR 97005
Phone 503-626-6877

Encounter REVELATION

For $950, Revelation Software by Cosmos will give your PC capabilities that no microcoputer has ever had before. Through a versatile, high performance Relational Data Base Management System developed by Cosmos, Revelation will give your microcomputer all the power and sophistication of a minicomputer.

Revelation features a data base with variable-length fields and records, (up to 64K), and unlimited files and accounts. File size is limited only by disk size.

Also featured: A powerful, hybrid programming language.

A fourth generation applications and program generator Our programs write programs!

Communications functions that give your PC the power to utilize applications, programs, and data from thousands of existing mainframe and minicomputers.

8087 math chip support

Revelation co-exists with MS/DOS™ or PC/DOS™, 1.0, 1.1, 2.0 versions, and is compatible with the IBM PC™, XT™, COMPAQ™, Corona™ Columbia Multi-Personal® and the Eagle 1600™. It's also compatible with minicomputers utilizing the PICK Operating System.™ Minimum configuration requires 320K memory and 8087 math chip.

MS/DOS™ of Microsoft Corp. IBM PC Registered TM of International Business Machines Corp. COMPAQ™ of COMPAQ Computer Corp. PICK Operating System™ of PICK SYSTEMS. Eagle 1600™ of Eagle Computers. Corona PC™ of Corona Data Systems. Columbia Multi-Personal™ of Columbia Data Products Inc.

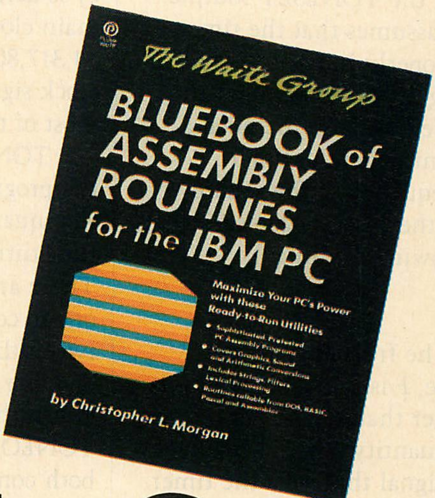
CIRCLE NO. 132 ON READER SERVICE CARD

COSMOS

P. O. Box AH, Morton, WA 98356

Telephone 1-800-422-2511
Inside Washington 206-496-5974.

Assembly Language Routines To Control Sounds On The IBM PC



An excerpt from Bluebook of Assembly Routines for the IBM PC

Editor's Note: This is the second excerpt from the forthcoming Bluebook of Assembly Routines for the IBM PC, *a Waite Group book by Christopher L. Morgan. (Morgan is also the author, with Mitchell Waite, of* 8086/8088 16-Bit Microprocessor Primer *and* Graphics Primer for the IBM PC.) *This book is part of the New American Library series of IBM PC computer-language books issued under the Plume/Waite imprint.*

The first excerpt, which appeared in the March issue of PC Tech Journal, *concerned assembly language routines to control the graphics capabilities of the IBM PC.*

Christopher L. Morgan

Sound is an important component in making computers behave in a more friendly manner. Right now, sound is more important in games than in other areas of computing; however, once computers are able to accurately and efficiently understand spoken commands, sound will most likely be one of the primary methods of input and output.

The routines that follow allow the programmer to use the speaker on the IBM PC to produce various sounds and musical tones. The first four routines—TONEINIT, TONESET, TONEON, and TONEOFF—are primitive functions; they access the speaker and its associated timer directly. All other routines must call these routines to produce sound.

The speaker is connected to the output of the timer, which produces a square wave whose frequency can

be set using the TONESET routine. TONESET assumes that the timer has been properly initialized. This is done during normal boot-up of the IBM PC. We have also included a routine to initialize the timer.

The frequency of the square wave from the timer is determined by the following formula:

$$f = F/n$$

where f is the frequency of the square wave, F is 1,193,182, and n is a 16-bit integer that is input to the routine. The quantity F is the frequency of a clock signal that runs the timer. This frequency is exactly one-third the frequency of the NTSC subcarrier used for color encoding by the color/graphics adapter. The frequency is derived by hardware from a main clock signal that runs at 14,317,800 cycles per second. This clock signal supplies the timing for most of the computer.

TONEON and TONEOFF give the programmer control over whether the square wave reaches the speaker, thus turning on and off the sound. There are actually two bits involved, one to control the connection between the clock signal and the timer and one to control the connection between the timer and the speaker. TONEON and TONEOFF switch both connections simultaneously. We found it unnecessary to control the bits independently.

The routines DELAY, FREQ, and TONE are used for making tones. DELAY provides timing for the duration of musical tones in milliseconds, and the routine FREQ uses the formula

$$n = F/f$$

to compute the input parameter for TONESET from a given frequency. Use this just before the TONESET routine to work directly with frequencies rather than with clock cycles of the 1,193,182-hertz clock.

TONE uses the other routines to produce tones of a given frequency and duration. The frequency is input as a 16-bit integer, and the duration is provided in milliseconds using the DELAY routine.

LISTING 1: TONEINIT—INITIALIZE SPEAKER TIMER

TONEINIT
Initialize Speaker Timer
FUNCTION: This routine initializes the portion of the 8253 timer chip used by the speaker system. In particular, it sets up channel 2 of this timer as a square-wave generator. This routine neither selects the frequency nor turns on the tone.
INPUT: None
OUTPUT: Only to timer 2 of the speaker circuit
REGISTERS USED: No registers are modified.
SEGMENTS REFERENCED: None
ROUTINES CALLED: None
SPECIAL NOTES: None
CODE:

```
; ROUTINE TO SET TONE;
; define control bit field parameters for the timer chip
sc      =       2                 ; use counter 2
rl      =       3                 ; mode to load period one byte
                                  ; at a time
mode    =       3                 ; square wave generator
bcd     =       0                 ; not bcd, use binary values;
; form control word
cnword  =       sc*40h + rl*10h + mode*2 + bcd
;
toneinit        proc    far;
        push    ax                ; save registers
; send control word to 8253 timer chip
        mov     al,cnword         ; select the above control word
        out     43h,al            ; send it out the control port
;       pop     ax                ; restore registers
        ret
;
toneinit        endp
```

LISTING 2: TONESET—SET THE TONE ON THE SPEAKER

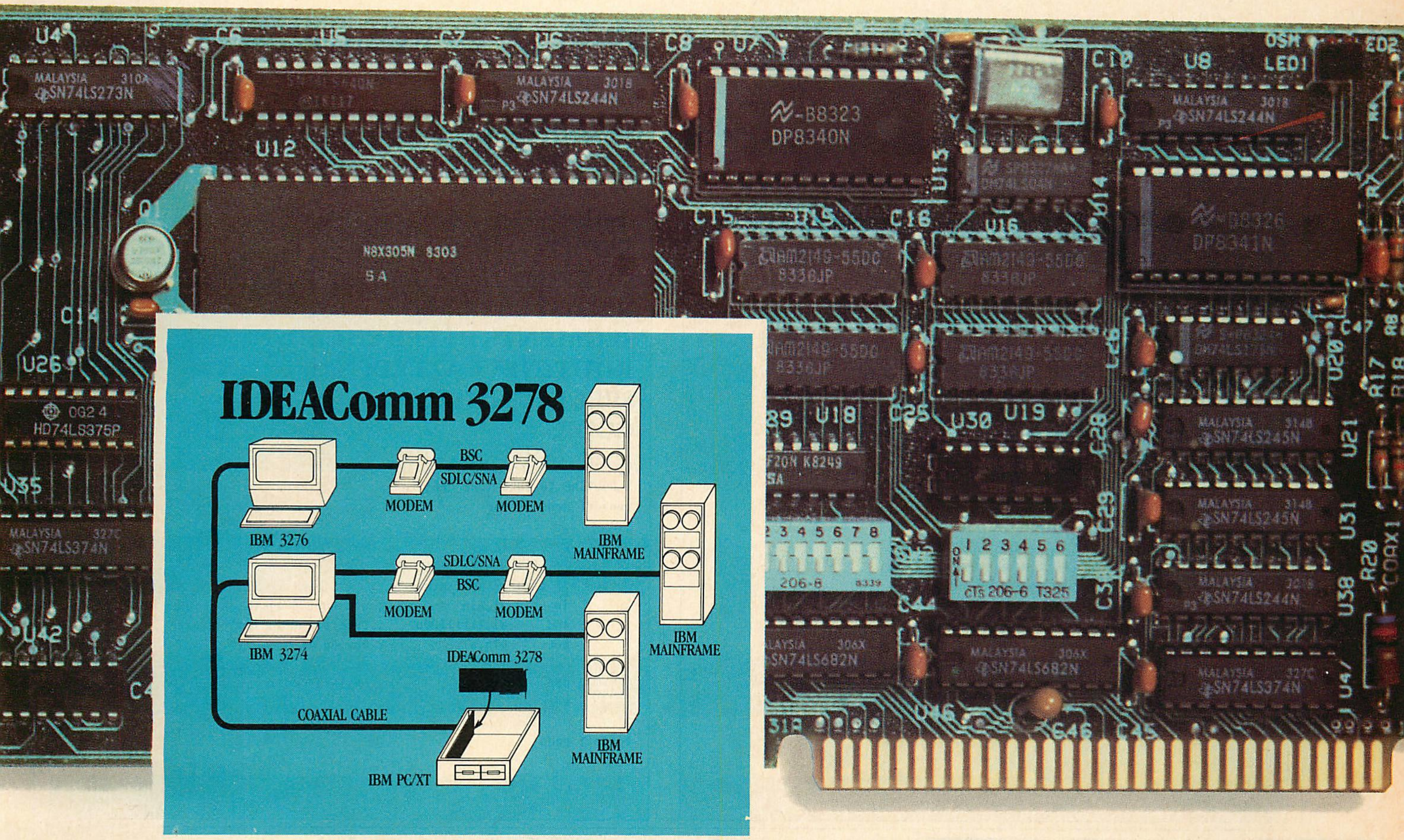
TONESET
Set the Tone on the Speaker
FUNCTION: This routine selects the frequency of the square-wave tone to the speaker. The input to this routine is a 16-bit integer n that determines the frequency f according to the following formula:

$$f = F/n$$

where F is 1,193,182, the frequency of a clock signal that feeds the timer. The value n is the number of cycles of the clock signal per cycle of the resulting square wave. This routine assumes that the speaker timer has already been properly initialized.
INPUT: Upon entry the 16-bit integer n is in the CX register.
OUTPUT: Only to timer 2
REGISTERS USED: No registers are modified.
SEGMENTS REFERENCED: None
ROUTINES CALLED: None
SPECIAL NOTES: None
CODE:

```
; ROUTINE TO SELECT TONE;
toneset proc    far
;
        push    ax              ; save registers
; load the time period into the timer
        mov     al,cl           ; lower byte
        out     42h,al          ; out to timer
        mov     al,ch           ; upper byte
        out     42h,al          ; out to timer
;
        pop     ax              ; restore registers
        ret
;
toneset endp
```

THE WORD IS OUT FROM IDE:



"Our IDEAComm 3278 for IBM PC to mainframe communications is only part of the solution."

At IDE, we know you don't communicate in a vacuum. If you are thinking of linking your PC to a mainframe, your application probably has many other requirements as well: extra memory, disks, modems, other peripheral devices. You need the assurance that *all* these products will work together to completely satisfy your application needs.

Of course, our IDEAComm 3278 has all the key features: Bisync and SNA/SDLC support • capture and storage of screen formats • uploading/downloading files • remote printer support • software reconfiguration of the keyboard • high speed transmission.

But as we said, this is only part of the solution. The rest of the solution is in the rest of the IDEAline – IDEAplus multifunction boards, IDEAdisk Winchester disk drives, IDEAComm 1200 integral modems, IDEAshare and IDEAnet resource sharing/networking products, IDEAgraph high resolution color graphics.

IDE gives you the true solution: broad line support from one vendor, one year product warranty, and a unique upgrade policy that lets you trade up any product for another in the line. Ask your computer retailer or call us at **800-257-5027** (in Massachusetts, call (617) 275-4430). We'll get the word out to you on IDE.

IDE Associates

Better ideas for personal computers.

IDE Associates, Inc., 7 Oak Park Drive, Bedford, MA 01730.

LISTING 3: TONEON—TURN ON TONE

TONEON
Turn on Tone
FUNCTION: Turns on the timer and speaker to produce a tone. The frequency of the tone must have already been selected on the timer.
INPUT: None
OUTPUT: To the timer and speaker only
REGISTERS USED: No registers are modified.
SEGMENTS REFERENCED: None
ROUTINES CALLED: None
SPECIAL NOTES: None
CODE:

```
; ROUTINE TO TURN ON TONE
;
toneon  proc    far
;
        push    ax                      ; save registers
;
; turn speaker and timer on
        in      al,61h                  ; get contents of system port
                                        ; B
        or      al,3                    ; turn speaker and timer on
        out     61h,al                  ; send out new values to port
                                        ; B
;
        pop     ax                      ; restore registers
        ret
;
toneon  endp
```

LISTING 4: TONEOFF—TURN OFF TONE

TONEOFF
Turn off Tone
FUNCTION: This routine turns off the timer and speaker.
INPUT: None
OUTPUT: To the timer and speaker only
REGISTERS USED: No registers are modified.
SEGMENTS REFERENCED: None
ROUTINES CALLED: None
SPECIAL NOTES: None
CODE:

```
; ROUTINE TO TURN TONE OFF;
toneoff proc    far
;
        push    ax                      ; save registers
;
; turn off timer 2 and speaker
        in      al,61h                  ; get port B again
        and     al,11111100b            ; turn off timer and speaker
        out     61h,al                  ; now do it
;
        pop     ax                      ; restore registers
        ret
;
toneoff endp
```

LISTING 5: DELAY—DELAY FOR SPECIFIED TIME INTERVAL

DELAY
Delay for Specified Number of Milliseconds
FUNCTION: Delays a specified number of milliseconds.
INPUT: Upon input CX contains the number of milliseconds to delay.
OUTPUT: None
REGISTERS USED: No registers are modified.
SEGMENTS REFERENCED: None
ROUTINES CALLED: None
SPECIAL NOTES: None
CODE:

```
; ROUTINE TO DELAY SPECIFIED NUMBER OF MILLISECONDS
delay   proc    far
;
        push    cx              ; save registers
;
delay1:
        push    cx              ; save counter
        mov     cx,260          ; timing constant
delay2:
        loop    delay2          ; small loop
        pop     cx              ; restore counter
        loop    delay1          ; loop to count milliseconds
;
        pop     cx              ; restore registers
        ret
;
delay   endp
```

LISTING 6: FREQ—CONVERSION FROM FREQUENCY TO PERIOD

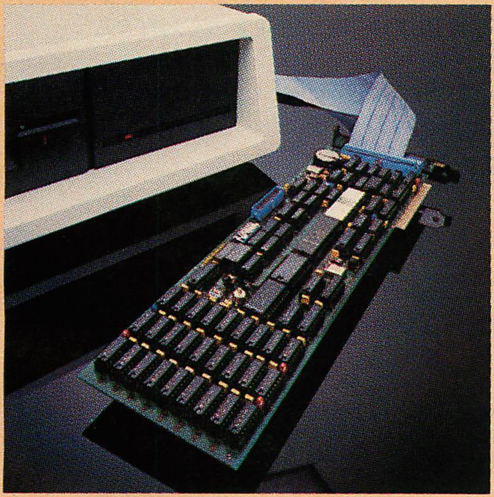
FREQ
Conversion from Frequency to Period
FUNCTION: This routine converts from frequency to the number required by TONESET to set the frequency. The routine uses the following formula:

$$n = F/f$$

where f is the frequency input to this routine, n is the number output by this routine, and F is 1,193,182. In other words, this routine divides the specified frequency f into the clock frequency of 1,193,182 hertz that drives the timer. Use this routine just before TONESET.
INPUT: Upon entry the frequency is in CX.
OUTPUT: Upon exit F/f is in CX.
REGISTERS USED: Only CX is modified.
SEGMENTS REFERENCED: None
ROUTINES CALLED: None

10 Mb more for your IBM PC.
6 times faster than an XT.
Plus: A serial port
A clock/calendar
Sockets for 192K of RAM
A hard disk controller
In just one expansion slot!
...and $500 less.

Now you can get 10 or 15 Megabytes of mass storage AND the most popular system expansion functions for your IBM PC at about $500 less than you'd pay for an IBM XT. Introducing the Falcon PC e**XT**ender:™ the fastest hard-disk system for the IBM PC.



Time=money: Save both!

The PC e**XT**ender offers more on-line capacity than 45 floppy disks eliminating time-consuming disk changes. Tim Paterson, author of the operating system for the IBM PC, has developed advanced circuitry that continually transfers information at the maximum speed of the drive. It can read one megabyte from the disk every three seconds. That's more than an entire floppy disk per second!

Save space too!

In just one expansion slot, the PC e**XT**ender features a multi-function hard disk controller board, a clock/calendar, a serial port (for use with a modem or printer), and space for 192K of RAM expansion. This economical arrangement leaves room to add additional capabilities to your PC in the future.

An emphasis on quality

Requiring no external power supply, the PC e**XT**ender has been designed to be an extraordinarily reliable system. It's rugged construction is backed by a full one-year warranty. For service you can simply return the unit to your dealer or to Falcon Technology. Should you ever have a problem with your PC e**XT**ender, Falcon has a customer service engineer available by phone to answer your questions.

Stretch your capabilities, not your budget

Prices for Falcon PC e**XT**ender systems start at $2295 for 10Mb mounted within the IBM chassis. To take your IBM PC beyond even an XT, the 15Mb drive sells for $2595. The PC e**XT**ender systems are also available in externally-mounted cases. Systems can easily be configured with either one or two Falcon drives.

Turn your PC into something even better than an XT.
The new Falcon PC eXTender does more, faster, for less.

FALCON

Falcon Technology, Inc.
6644 S. 196th St., Suite T-101
Kent, WA 98032
1-800-722-2510
(In Washington state)
(206) 251-8282

CIRCLE NO. 170 ON READER SERVICE CARD

IBM is a registered trademark of the International Business Machines Corporation.

SOUND EXCERPT

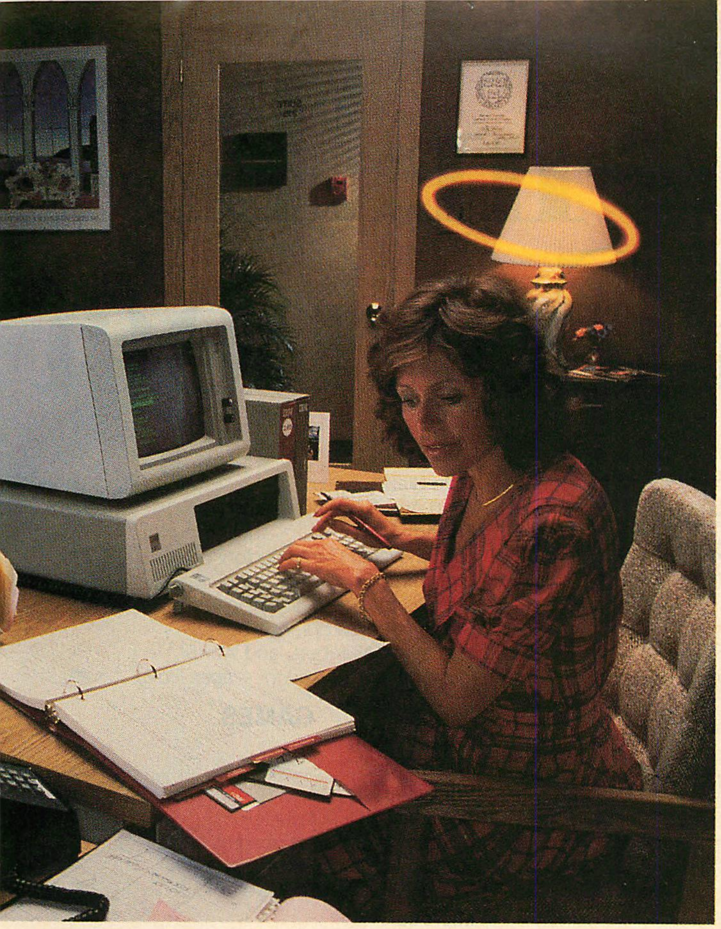
SPECIAL NOTES: None
CODE:

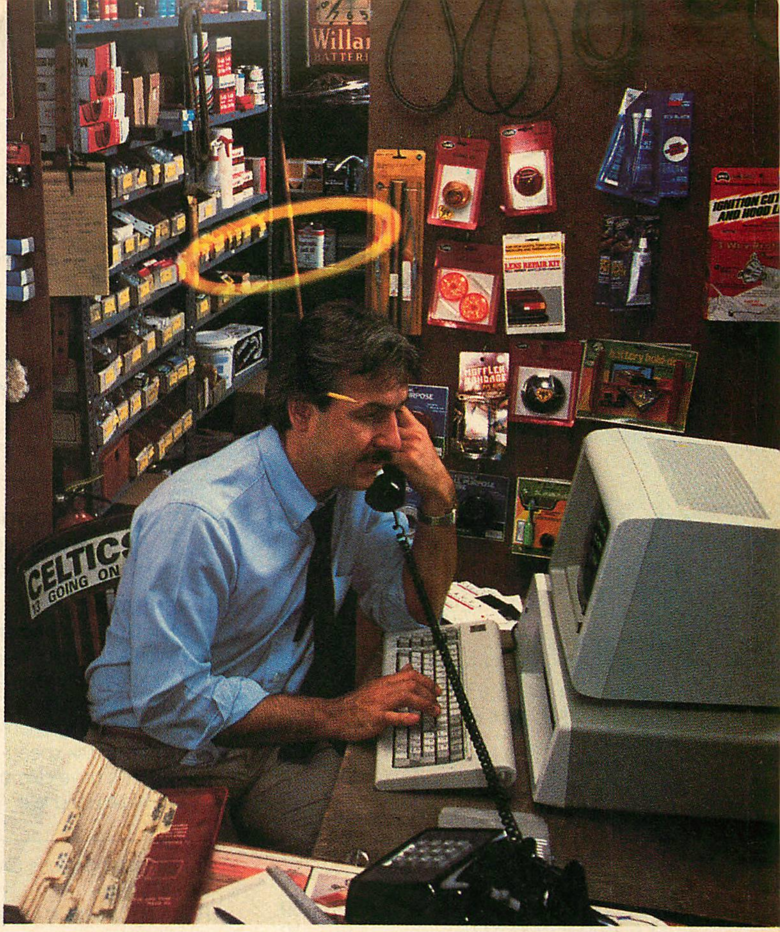
```
; ROUTINE TO CONVERT FROM FREQUENCY TO PERIOD
;
freq    proc    far
;
        push    dx              ; save registers
        push    ax
;
        mov     dx,12h          ; upper part of numerator
        mov     ax,34DEh        ; lower part of numerator
        div     cx              ; divide by frequency
        mov     cx,ax           ; the quotient is the output
;
        pop     ax              ; restore registers
        pop     dx
        ret
;
freq    endp
```

LISTING 7: TONE—MAKE A TONE

TONE
Make a Tone
FUNCTION: This routine makes a tone of a given frequency and length.
INPUT: Upon entry the frequency is in CX and the length in number of milliseconds is in DX.
OUTPUT: To the speaker and timer only
REGISTERS USED: No registers are modified.
SEGMENTS REFERENCED: The data segment must contain the variable COUNT.
ROUTINES CALLED: TONESET, TONEON, TONEOFF, DELAY
SPECIAL NOTES: The speaker timer must already have been properly initialized. This should happen during boot-up.
CODE:

```
; ROUTINE TO MAKE TONE
;
tone    proc    far
;
        push    dx              ; save registers
        push    cx
        push    ax
;
; compute the frequency and set up the tone
        call    freq            ; convert the frequency
        call    toneset         ; set up the tone
;
; turn on the tone
        call    toneon          ; turn it on
;
; wait for proper delay
        mov     cx,dx           ; get delay length
        call    delay
;
; turn off the tone
        call    toneoff         ; turn it off
;
        pop     ax              ; restore registers
        pop     cx
        pop     dx
        ret
;
tone    endp
```





Divine inspiration for people who use the IBM PC, XT and PCjr.

Introducing SAVVY PC™. Now you can create custom databases in less time, with less effort and for less money.

SAVVY PC is unique. It integrates a database management system with a truly "natural" command language and a self-documentation capability. There has never been a faster or easier database building tool than SAVVY PC.

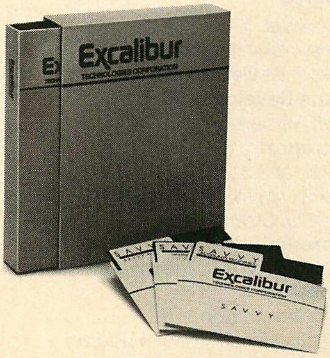
SAVVY PC speaks English. You won't have to master the rigid syntax and complex symbols of typical computer programming languages to develop a wide range of business and personal applications.

Because SAVVY PC is a fully integrated database, your applications can directly access all master files, speeding and simplifying operations. And because SAVVY® can recognize your personal word patterns, you can misspell, mistype, abbreviate or rephrase and still command your computer.

The speed, flexibility and economy of computing with SAVVY PC is truly inspired.

For more information and the name of the SAVVY dealer in your area, or to arrange for a SAVVY PC demonstration call 1-800-551-5199. (In New Mexico call 505-242-3333.)

SAVVY PC is a product of Excalibur Technologies Corporation, 800 Rio Grande Boulevard N.W., Mercado 21, Albuquerque, New Mexico 87104.



Dealer Inquiries Invited

SAVVY®

A product of **Excalibur**
TECHNOLOGIES CORPORATION

CIRCLE NO. 267 ON READER SERVICE CARD

In IBM
In PC Connect

SOFTWARE

ApTec *(for color Prism Printers)*
Rainbow Writer Color Text Formatter . . . $119.
Rainbow Writer Screen Grabber . . . . 69.
Ashton Tate
dBase II . . . . 379.
Best Programs
PC/Personal Finance Program . . . . 65.
PC/Professional Finance Program II . . . . 169.
PC/Fixed Asset System . . . . 279.
PC/Tax Cut . . . . 175.
Bible Research
THE WORD *(KJV Bible — 7 disks)* . . . . . 145.
Conceptual Instruments
The Desk Organizer *w/ 1 yr. free updates & 30 day return guarantee* . . . . 197.
Continental
Home Accountant Plus . . . . 89.
FCM *(was First Class Mail)* . . . . 79.
UltraFile *(file/report/graph)* . . . . 119.
Digital Research
CP/M-86 . . . . 39.
Dr.LOGO . . . . 79.
Financier
Financier II *(was Personal Series)* . . . . 117.
Financier Tax Series . . . . 97.
Funk Software
Sideways . . . . 49.
IUS
EasyWriter II System
EasyWriter II, EasySpeller II, EasyMailer II 239.
EasySpeller II *for EasyWriter II* . . . . 59.
EasyMailer II *for EasyWriter II* . . . . 59.

Accounts Receivable . . . . 309.
Accounts Payable . . . . 309.
General Ledger . . . . 309.
Inventory . . . . 309.
Order Entry . . . . 309.
Package Price for any three above . . . . . 859.
Payroll . . . . 379.
Lifetree
Volkswriter . . . . 115.
Volkswriter International . . . . 135.
Volkswriter Deluxe *(with TextMerge)* . . . . . 175.
Lotus Development
1-2-3 *(version 1A)* . . . . call
Micropro
WordStar 3.3 . . . . 269.
ProPak *(WordStar/MailMerge/SpellStar/StarIndex)* . . . . 379.
Microrim
R:base 4000 *(fully relational database)* . . . 329.
Microsoft
Multiplan *(Version 1.1)* . . . . 159.
Multi-Tool Financial *(for Multiplan)* . . . . 69.
Multi-Tool Budget *(for Multiplan)* . . . . 99.
Multi-Tool Word . . . . 249.
Multi-Tool Word *(with mouse)* . . . . 297.
Microstuf
Crosstalk XVI . . . . 109.

PC Connection Software Special
through April 30, 1984

VERBATIM
Disk Drive Analyzer
- Works on floppy drives only
- Automatically tests 4 critical areas of drive performance
  1) Radial Alignment test
  2) Tests disk speed
  3) Tests drive clamping accuracy
  4) Write/Read test **$29**

Transporter *(includes Crosstalk)* . . . . . . $169.
PCsoftware
PCrayon *(create in full color)* . . . . 34.
Peachtree
PeachText 5000 *(now with ATI Training)* . . 207.
Peter Norton
Norton Utilities . . . . 55.
Rosesoft
ProKey 3.0 *(new version/new features)* . . . 95.
Software Arts
TK!Solver . . . . 189.
Software Products
Open Access . . . . 399.
Software Publishing
PFS/File . . . . 95.
PFS/Graph . . . . 95.
PFS/Write . . . . 95.
PFS/Report . . . . 79.
Softword Systems
Multimate *(w/spelling checker)* . . . . call
Virtual Combinatics
Micro Cookbook . . . . 29.
VisiCorp
VisiCalc IV *(w/StretchCalc)* . . . . 159.
VisiWord Plus *(with VisiSpell)* . . . . 259.
StretchCalc *(for VisiCalc)* . . . . 75.

TRAINING

ATI
How to use Lotus 1-2-3 . . . . 59.
How to use Wordstar *(vol 1 & 2)* . . . . 59.
How to use dBase II *(vol 1 & 2)* . . . . 59.
How to use EasyWriter II . . . . 59.
How to use Multiplan . . . . 59.
How to use Your IBM-PC . . . . 59.
How to use Multimate . . . . 59.
How to use Microsoft Word . . . . 59.
How to use PC DOS . . . . 59.
How to use Peachtree A/R, A/P, or G/L *(specify)* . . . . 59.
Orion
Hello Charlie . . . . 75.

EDUCATIONAL

Davidson
Speed Reader II . . . . 49.
Math Blaster *(ages 6 to 12)* . . . . $35.
Word Attack *(ages 8 to adult)* . . . . 35.
FriendlySoft
FriendlyWare/PC Introductory Set . . . . 39.
Lightning Software
Master Type *(the best typing tutor!)* . . . . . . 35.
Stone *(requires graphics board)*
My Letters, Numbers, and Words
(great graphics & sound — ages l to 5) . . . 29.

GAMES

Friendly Soft
FriendlyWare/PC Arcade . . . . 39.
Funtastic
Snack Attack II *(a favorite)* . . . . 27.
Cosmic Crusader *(as good as above)* . . . . 27.
Big Top *(climb to new levels)* . . . . 29.
Master Miner *(1 or 2 players)* . . . . 29.
Microsoft
Flight Simulator *(B & W on RGB monitors)* 35.
Orion
J-Bird (you'll never sleep!) . . . . 29.
PC-MAN . . . . 27.
Paratrooper . . . . 25.
Pits & Stones . . . . 29.
PCsoftware
Armchair Quarterback . . . . 27.
Chess 2001 . . . . 27.
Sierra/On-Line
Frogger . . . . 27.
Crossfire *(keyboard or joystick)* . . . . 23.
Ulysses and the Golden Fleece . . . . 27.
Sublogic
Night Mission Pinball . . . . 29.

HARDWARE

AST Research *(For IBM-PC or XT)*
All AST Boards come with SuperDrive, SuperSpool, and one year warranty.
SixPakPlus 64k upgradable to 384k, with clock calendar, serial and parallel ports *(game port optional)* . . . . 269.
MegaPlus II 64k upgradable to 256k (or more with MegaPak) with clock calendar and serial port *(parallel, game, or second serial port optional)* . . . . 269.
MegaPak 128k *(not upgradable)* . . . . 187.
MegaPak 256k . . . . 279.
I/O Plus II with clock calendar and serial port *(parallel, game, or second serial port optional)* . . . . 115.
Parallel, Game, or second Serial Port *for any AST board (specify board)* . . . . 35.
Connectall *connector bracket (PC only)* . . . 15.
Amdek
Video 300G Monitor *(green)* . . . . 149.
Video 300A Monitor *(amber)* . . . . 159.
MAI Board *(now w/Halo software)* . . . . . . 469.
CompuCable
Keyboard & Drive Cover set . . . . . see special
IBM Mono Screen Enhancement see special

*DEFECTIVE SOFTWARE REPLACED IMMEDIATELY DEFECTIVE HARDWARE REPLACED OR REPAIRED AT OUR DISCRETION SOME ITEMS HAVE WARRANTIES UP TO FIVE YEARS

you trust.
ion you believe.

Printer to IBM Cable *(specify printer)* .... $32.
Smartmodem to IBM Cable .... 25.
Curtis
NEW Low Profile Tilt & Swivel Pedestal
(for any monitor) .... 44.
PC Pedestal *(for IBM Mono or Color)* ..... 59.
PGS or Quadchrome display adapter ..... 9.
System Stand *(holds PC vertically)* .... 21.
Extension Cables for IBM Mono Display .. 44.
Keyboard Extension Cable *(3 to 9 feet)* ... 36.
AC Plug Adapter
(any monitor to your PC) .... 8.
Electronic Protection Devices
All are rated at 1875 watts, 15 AMP.
The Lemon .... 39.
The Lime .... 59.
The Peach .... 64.
The Orange .... 94.
Epson
RX-80 with GRAFTRAX-Plus .... call
RX-80 F/T with GRAFTRAX-Plus .... call
FX-80 with GRAFTRAX-Plus .... call
FX-100 with GRAFTRAX-Plus .... call
Epson to IBM Parallel Cable .... 32.
Hayes
Smartmodem 300 .... 209.
Smartmodem 1200 .... 499.
Smartmodem 1200B .... 419.
Smartcom II .... 99.
Smartmodem-to-IBM Cable .... 25.
Hercules Computer
Hercules Graphics Card *(parallel port)* ... 349.
IDS
Prism 80 Printer *(with all four options)* .. 1397.
Prism 132 Printer *(with all four options)* 1597.
Prism to IBM Parallel Cable .... 32.
Koala
Koala Touch Tablet *with software*
(connects to game port) .... 97.
Kraft
Joystick .... 39.
Game Paddles *(pair)* .... 29.
Maynard Electronics
Floppy Drive Controller .... 179.
Internal Hard Disk (10 Meg), WS-1 Controller
& ROM for PC mother board .... 1079.
Mouse Systems
PC Mouse *(3 button optical mouse comes with power supply, desk pad, and software — connects to serial port)* .... 239.
NEC
Spinwriter 3550 *(IBM-PC compatible)* .. 1799.
Spinwriter 2050 *(3550's little brother)* .... 995.
Paradise Systems
MultiDisplay Card *(color & mono)* ...... 397.
Plantronics/Frederick
COLORPLUS *(now w/Color Magic)* .... call
Princeton Graphics
HX-12 RGB monitor *(690 x 240)* .... call

PC Connection Hardware Special
through April 30, 1984

COMPUCABLE
Plastic Keyboard and Drive Cover Set
Smoke colored, hard plastic to protect your PC or XT **$13**
Screen Enhancement for IBM Monochrome Screen
Snug fitting cover cuts screen glare without any loss of character quality . **$13**

CONNECT WITH US

At PC Connection, we know how hard it can be to find the ideal solutions to your PC needs. And with the constant barrage of new products and new technology, the best configuration yesterday might be less-than-ideal today. Keeping up with the world of IBM Personal Computers is a full-time job. At PC Connection it's our only job. Our trained support staff will gladly take the time to discuss your system and your requirements. Call us anytime Monday through Friday 9:00 to 9:00, or Saturday 9:00 to 6:00, at 603/446-3383.

OUR POLICY

- We accept VISA and MASTERCARD.
- No surcharge added for charge cards
- Your card is not charged until we ship.
- No sales tax.
- All shipments insured; no additional charge.
- Allow 1 week for personal & company checks to clear.
- COD max. $500. Cash or certified check.
- 120 day guarantee on all products.*

SHIPPING

Continental US: For monitors, printers, and drives, add 2% to all orders. For all other items, add $2 per order for UPS surface, $3 per order for UPS 2nd-Day-Air.
Outside Continental US: We add freight charges to credit card purchases. For prepayment, call 603/446-3383 for information.

800/243-8088

PC Connection
6 Mill St., Marlow, NH 03456
603/446-3383

For the IBM-PC *Exclusively.*

CIRCLE NO. 182 ON READER SERVICE CARD

MAX-12 Amber monochrome monitor ... call
Quadram
We are a full line Quadram Dealer
New Expanded Quadboard 64k expandable to 384k, with clock calendar, parallel, serial & game port, I/O bracket, and Quadmaster software .... $269.
Microfazer Printer Buffer (parallel) w/copy
MP 64 (64k) *upgradable to 512k* ...... 197.
Quadcolor I .... 197.
Quadlink *(allows you to run most Apple II programs directly on your IBM-PC or XT)* 459.
Quadisk *(various sizes to 72 meg)* ...... call
STB
RIOPlus 64k *(upgradable to 384k) with PC accelerator, clock calendar, serial and parallel port, and "Connectall" type bracket (game cable optional)* .... 259.
Graphix Plus *(color & mono)* .... 379.
Silver Reed *(letter quality, 132 column)*
Silver Reed EXP 550 Printer (16 CPS) ... 589.
Silver Reed EXP 770 Printer *(30 CPS)* ... 950.
TG
Joystick .... 45.
Tandon *All drives are competely pre-tested*
TM 100-2 (5 1/4") full-height drive (DS,DD)
Specify Drive A or Drive B for your PC
Comes with complete step by step installation instructions .... 239.
USI *(monitors for graphics board)*
Pi-2 Monitor (12" green, with cable) .... 149.
Pi-3 Monitor (12" amber, with cable) .... 159.

MEMORY

64k Memory Upgrade Set for IBM-PC or XT system board .... 59.
64k Memory Upgrade Set for any memory board *specify make of board* .... 59.
Install memory upgrades & run diagnostics *at time of board purchase only* .... 10.

DISKS

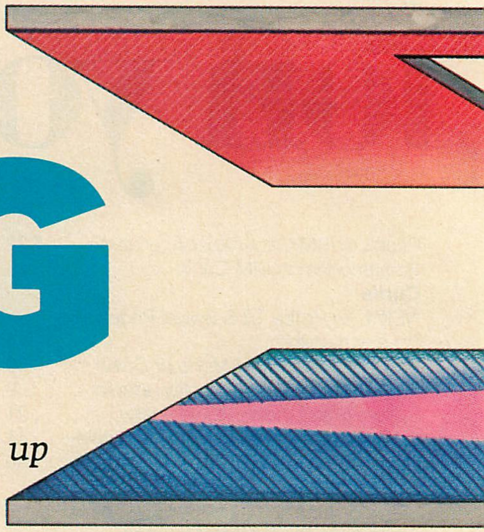
Verbatim *(with 5 year guarantee)*
Datalife Disks SS/DD *plastic box of 10* .... 29.
Datalife Disks DS/DD *plastic box of 10* ... 39.
Flip Sort *(holds 75 disks)* .... 19.

INFORMATION SERVICES

Compuserve
Compuserve Information Service *includes subscription, manual, 5 hours of connect time, & monthly publications)* .... 39.
Executive Information Service .... 79.
Vidtex Software .... 59.
Executive Information Service with Vidtex .... 119.
Source Telecomputing
The Source *(subscription & manual)* ..... 69.
The Source *with* Sourcelink Software .... 109.

PC CONNECTION™

MIXING SYSTEMS COMPONENTS

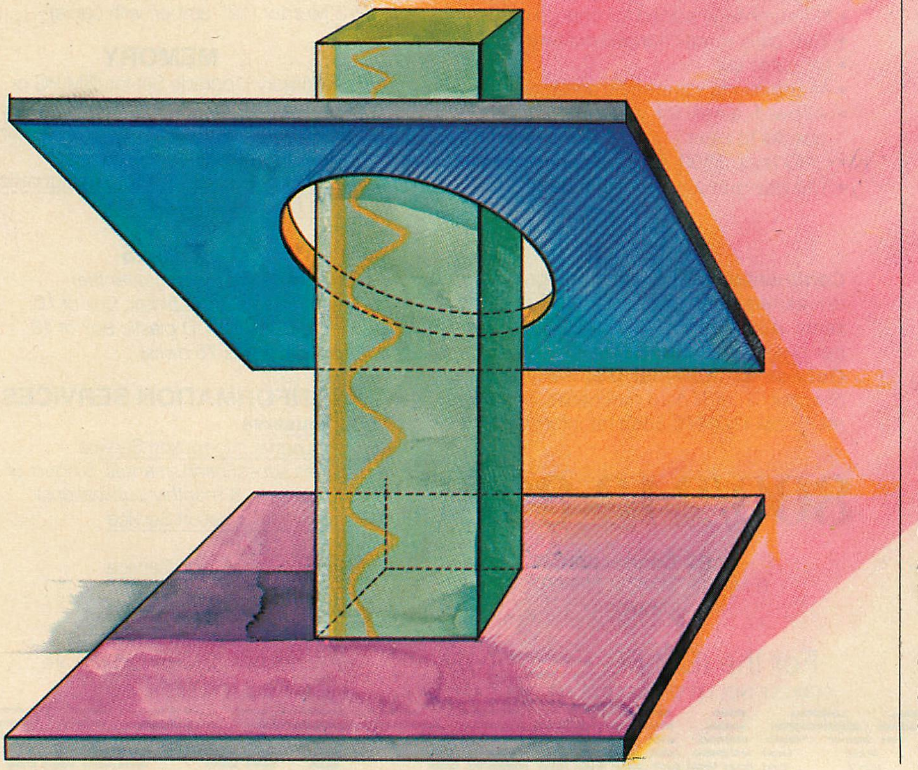


The Gemini Printer and GRAPHICS.COM had a disagreement over line spacing, but DEBUG patched things up

MORTON GOLDBERG

Anyone who has ever tried to combine systems components made by different companies knows that such implementations are often simpler in theory than they are in practice. Even when the components are supposed to be compatible, unexpected problems can develop. I was recently reminded of this when I tried to print out a graphics display using GRAPHICS.COM, an IBM Color/ Graphics Adapter, and a Star Micronics Gemini-10 matrix printer.

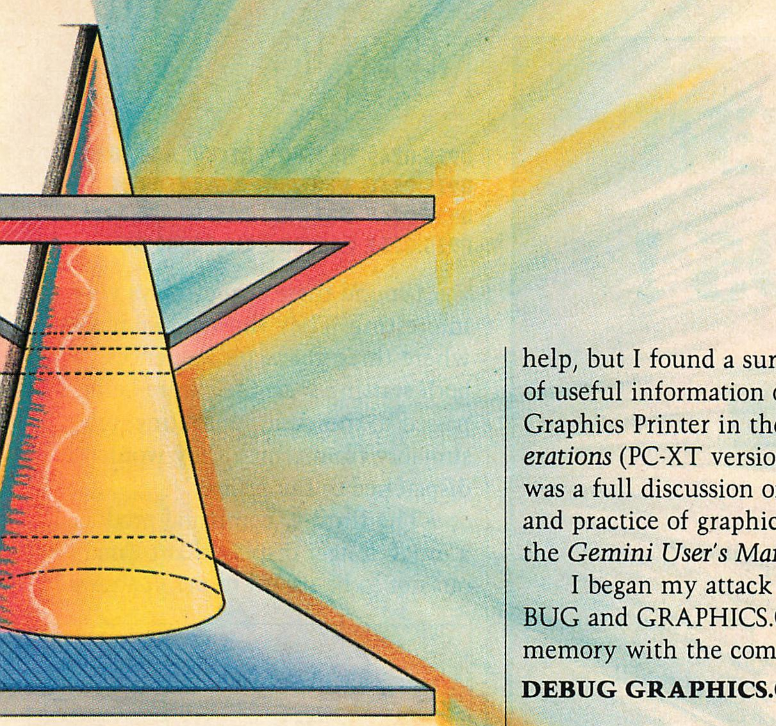
GRAPHICS.COM, a new utility included in PC DOS 2.0, extends the screen-print capability of the IBM PC or PC-XT. After running GRAPHICS.COM, which loads a graphics printer driver into memory, use the Shift-PrtSc key to print graphics displays as well as text displays. Printing text displays is much easier for PC-DOS than printing graphics, because to write software for printing text, the programmer need not assume anything special about the display adapter or the printer. In writing software for graphics displays, however, the programmer must be aware that GRAPHICS.COM requires both an IBM Color/Graphics Adapter and an IBM Graphics Printer or equivalent devices. Not every PC or PC-XT owner will have such equipment. This is undoubtedly why GRAPHICS.COM has been provided as a separate utility. If the purchaser can't use it, he doesn't load it, and it's probably not even on his system disk. This gives him a little more memory and a little more disk space than he would have if GRAPHICS.COM were integrated into COMMAND.COM.



I own an IBM Color/Graphics Adapter, but I don't own an IBM Graphics Printer. Instead, I have a Star Micronics Gemini-10 matrix printer. I didn't think this was going to cause me any trouble as far as GRAPHICS.COM was concerned. The dealer from whom I bought the Gemini told me that it was "100 percent Epson compatible." If that had been true, I wouldn't be writing this.

Although the Gemini's graphics controls and those used by Epson are similar, they are just different enough to mess things up. Of course, I didn't know this at first, so as soon

ILLUSTRATIONS • IZUMI INOUE



as I had a graphics display that I thought worthy of printing, I had no hesitation in calling on GRAPHICS.COM (via the Shift-PrtSc key) to do the job. Figure 1 shows how the resulting printout looked. Figure 2 shows what I was expecting.

I found this situation intensely frustrating, because it was only the

The dealer from whom I bought the Gemini told me that it was "100 percent Epson compatible." If that had been true, I wouldn't be writing this.

vertical line spacing that was wrong. Everything else was fine. Close, but no cigar. But if it was that close, perhaps a little patching could fix it. I don't believe Iwould have entertained this thought if GRAPHICS.COM had not been packaged as a separate utility. I would not have had the courage to mess around inside COMMAND.COM. Modularity in programs has great value, and not just to those who write the programs.

Before I could launch my attack on GRAPHICS.COM, I needed to gather as much knowledge as I could. The DOS 2.0 manual wasn't much help, but I found a surprising amount of useful information on the IBM Graphics Printer in the *Guide to Operations* (PC-XT version), and there was a full discussion on the theory and practice of graphics printing in the *Gemini User's Manual.*

I began my attack by loading DEBUG and GRAPHICS.COM into memory with the command

DEBUG GRAPHICS.COM

DEBUG, of course, was to be my main weapon in this attack.

I knew from reading the *Guide to Operations* that the IBM Graphics Printer needs a sequence of control codes starting with an ESC (1B hex) in order to set its vertical spacing for graphics. Somewhere GRAPHICS.COM must generate such a sequence. Because of this, I decided to search for occurrences of ESC codes, using the DEBUG S(earch) command.

S CS:0100 L315 1B

The numbers are hex. The command told DEBUG to search for ESC codes in 315 (hex) bytes of memory, starting at address CS:0100, that being the space occupied by GRAPHICS.COM.

It turned out, as much through luck as through foresight on my part, that searching for ESC codes was exactly the right thing to do. DEBUG's response to my search command was

```
0915:015D
0915:0244
0915:0305
```

This showed me that there were just three places in GRAPHICS.COM where ESC codes occurred. I had expected to find a half-dozen or more places and to spend a lot of time weeding out the spurious ones, so I was pleased to find only three.

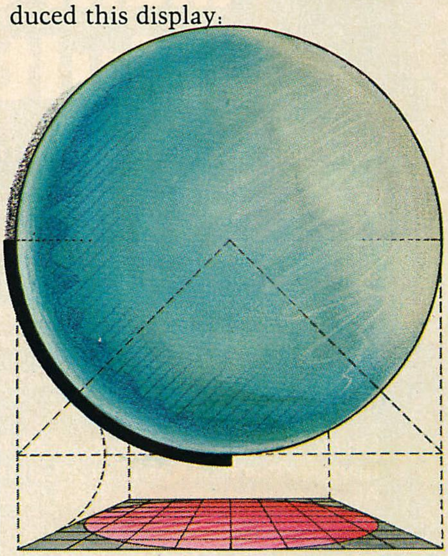
Next I had to disassemble the 8088 code in the neighborhood of the ESC codes. I used DEBUG's U(nassemble) command to do this. I had to try a couple of variations of the U command before I got the 8088 code to disassemble on proper instruction boundaries, but that's normal when you are disassembling code consisting of variable length instructions. The U commands that did the job were

```
U 015C L12
U 0243 L12
U 0301 L34
```

The first of these commands told DEBUG to disassemble 12 (hex) bytes of memory, starting at address CS:015C. This command produced a display that looked like this:

```
0915:015C B81B00   MOV AX,001B
0915:015F E87A01   CALL 02DC
0915:0162 B83300   MOV AX,0033
0915:0165 E87401   CALL 02DC
0915:0168 B81800   MOV AX,0018
0915:016B E86E01   CALL 02DC
```

The second U command, which differs only in its starting address, produced this display:



```
0915:0243 B81B00   MOV AX,001B
0915:0246 E89300   CALL 02DC
```

Morton Goldberg lives in Michigan and is a system administrator and programmer for a minicomputer-based Unix system. He also writes programs for Arborsoft, a one-man software house.

```
0915:0249 B83300    MOV AX,0033
0915:024C E88D00    CALL 02DC
0915:024F B82400    MOV AX,0024
0915:0252 E88700    CALL 02DC
```

I found these two displays very interesting. They showed two places where three-character sequences, both starting with ESC, were being passed to the same subroutine, presumably from which they would be dispatched to the printer.

The third U command produced a much longer display that turned out not to be useful to me. It showed

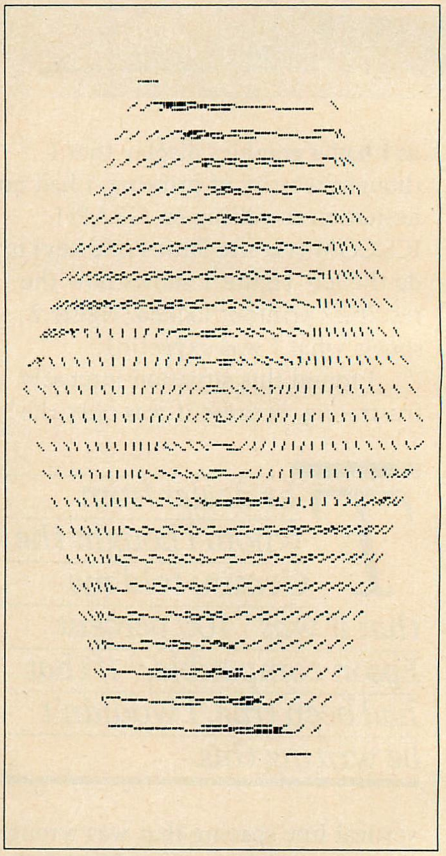


Figure 1: Before GRAPHICS.COM Was Patched

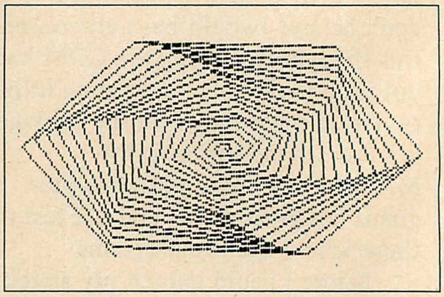
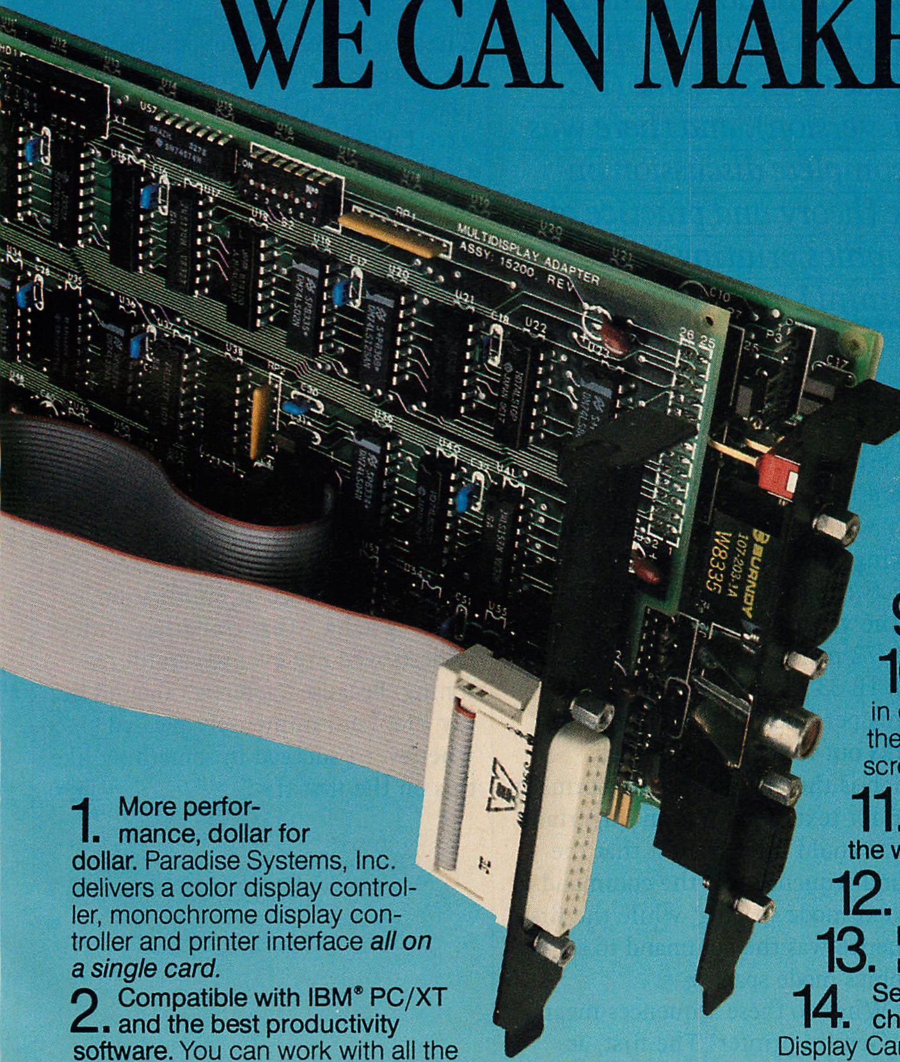


Figure 2: After GRAPHICS.COM Was Patched

14 REASONS WHY THE PARADISE MULTI-DISPLAY CARD IS SELLING FASTER THAN WE CAN MAKE THEM.



1. More performance, dollar for dollar. Paradise Systems, Inc. delivers a color display controller, monochrome display controller and printer interface *all on a single card.*

2. Compatible with IBM® PC/XT and the best productivity software. You can work with all the great IBM productivity packages, like Lotus 123™.

3. Standard monochrome display with no additional software: Fully IBM compatible. All the clean, fully-formed characters you'd expect from IBM.

4. High-resolution color graphics: Fully IBM compatible. 640 × 200 black and white; 320 × 200 medium resolution; 4 colors.

5. The best working display available for IBM. Flicker-free scrolling—even in color.

6. External switch for easy selection of monochrome or color mode on system startup without having to open the machine.

7. Switches easily between monochrome and color modes under software control.

8. IBM character sets and text modes. Paradise delivers character sets identical to IBM offerings. (9 × 14 character box for monochrome and 8 × 8 character box for color/graphic modes.)

9. Built-in fully compatible IBM type printer port.

10. Use two cards for simultaneous display. Two Multi-Display cards—or one in conjunction with an IBM video card—give you the ability to perform simultaneous display on two screens.

11. Individually selectable horizontal and vertical sync polarities: Compatible with the widest range of RGB monitors.

12. Illustrated manual with step-by-step instructions.

13. Fully IBM compatible light pen and RF modulator ports.

14. Separate connectors for easy function changes without changing jumpers. The Multi-Display Card has separate connectors for monochrome display; composite video; RGB video; parallel printer; light pen; and RF modulator.

It's no wonder the Multi-Display Card is in such high demand.

Look for the Multi-Display Card from Paradise Systems, Inc. at your local computer dealer. Call (415) 468-6000 or write to 150 North Hill Drive, Brisbane, CA 94005, for more information.

PARADISE
SYSTEMS, INC

what I construe to be the subroutine in which GRAPHICS.COM builds the sequence of control codes that it must put at the beginning of every line of graphics output it generates. It was far more likely that one of the first two disassemblies was the 8088 code that set the vertical spacing.

How could I be sure that I had located the right 8088 code? To find out, all I had to do was look up the sequences 1B 33 18 and 1B 33 24 in the printer section of the *Guide to Operations*. Sure enough, I found that they were commands to set vertical spacing. The first sequence, 1B 33 18, directs the IBM Graphics Printer to set the vertical spacing at 8 points per line (72 points = 1 inch) and the second, 1B 33 24, directs it to set 12 points per line. The latter setting works out to six lines per inch, which is the printer spacing normally used for text. That was an important clue. I could infer from it that the second sequence was the command to set text mode spacing, while the first sequence was the command to set graphics mode spacing.

What do these sequences mean to a Gemini printer? The first, according to the Gemini manual, directs it to set vertical spacing at 12 points per line, and the second directs it to set vertical spacing at 18 points per line. Not so good. Those big gaps in figure 1 were caused by a vertical line spacing of 24 points per line. I knew that because I had measured the gaps. Well, maybe the Gemini was double spacing. If that were true, the correct sequences for the Gemini would be 1B 33 08 (4 points per line, doubled) and 1B 33 18 (12 points per line). I decided to patch GRAPHICS.COM so that the Gemini would get these sequences rather than the ones it had been getting.

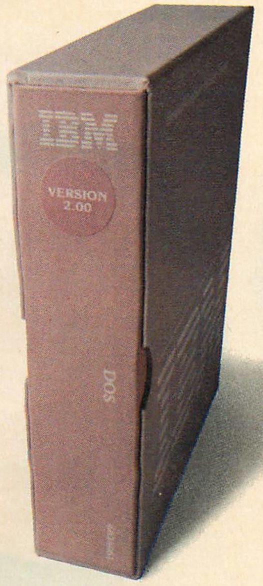
> *I found a surprising amount of useful information on the IBM Graphics Printer in the* Guide to Operations *(PC-XT version), and there was a complete discussion on the theory and practice of graphics printing in the* Gemini User's Manual.

I made the patches easily, using two DEBUG E(nter) commands, which looked like this:

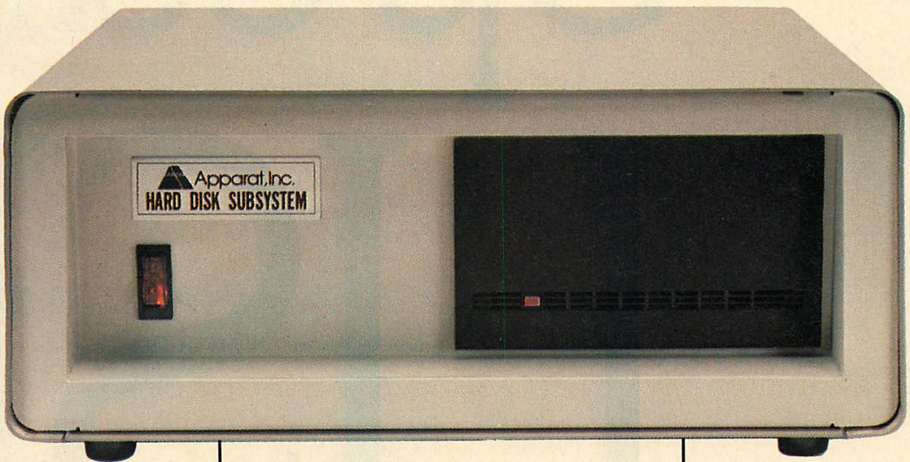
E CS:0169 08
E CS:0250 18

The two addresses, 0169 and 0250, were taken from the disassembly displays. Once the patches were made, I gave DEBUG a W(rite) command so that it would make a disk copy of my new GRAPHICS.COM, and then I said goodbye to DEBUG with the Q(uit) command.

After that there was nothing left to do but try it, so I did, and it worked just fine. Of course, you knew that it had, because if it hadn't, this wouldn't be the last paragraph. However, I wasn't confident that it would, because although I had been pretty careful in working out the patches, I might have overlooked some critical piece of information, some "gotcha," that could have utterly destroyed my ever-so-careful reasoning. Because "gotchas" abound in my life, I am immensely pleased whenever I succeed in something like this on the first try.



THIS IS ALL IT TAKES TO RUN OUR HARD DISK.



Compatibility. A lot of manufacturers talk about it. Mostly, compatibility is defined as something that almost works like the original. Our hard disk sub-system for the IBM PC runs on 2.0 without modifications, patches or use of device drivers. Just boot 2.0 and run.

And we haven't cut back on reliability either. the Apparat hard disk uses the Xebec controller and connects via a host interface module to one of the slots in your PC. The HIM also contains an RS-232 serial async port for use with a mouse, modem, etc. Even with compatibility and reliability built in, we've managed to provide some very competitive pricing. The 10, 15 and 26 megabyte drives are priced at **$2,295, $2,695** and **$3,395** respectively.

When you use a multifunction RAM Card, such as the Apparat Combo II (with up to 512K of RAM, RS232, parallel printer, clock calendar, and game adapter) along with the HIM, floppy disk controller and CRT monitor adapter cards in your PC, you still have one slot available for future expansion.

If you're considering a fixed disk for your IBM PC, look at all the systems available. Look for compatibility. Look for reliability. Look for value. We think you'll pick ours.

For more information or to order yours call **800/525-7674** or write Apparat, Inc., 4401 S. Tamarac Parkway, Denver, CO 80237, 303/741-1778. Dealer inquiries invited.

IBM PC is a registered trademark of International Business Machine Corp.

Price and specifications subject to change without notice.

Apparat, Inc.

CIRCLE NO. 106 ON READER SERVICE CARD

The PC Speaks LISP

Three respectable approaches to LISP on the PC: IQ LISP, TLC LISP, and muLISP

WILLIAM G. WONG

LISP (List Processing) is one of the main languages used by computer scientists working in the area of artificial intelligence (AI). It is also the oldest such language, predating most other computer languages (FORTRAN is one that may have been developed earlier). LISP was first developed by John McCarthy at Massachusetts Institute of Technology as a way to describe solutions to symbolic logic problems using Alonzo Church's lambda calculus; the original LISP 1.5 Programmer's Manual was made available in the early 1960s.

The language has been implemented on large minis and mainframes to satisfy the research needs of the AI community. Special LISP machines have even been made avail-

able in recent years by companies such as LISP Machines Incorporated (LMI), Symbolics, and Xerox. Much of the demand is due to the movement of AI techniques from universities to the marketplace.

Recent implementations of LISP on the IBM PC have made the power and flexibility of the language available to a large number of users at a reasonable cost. LISP systems usually require large amounts of memory to operate effectively; the IBM PC, with an address space of 1 megabyte, meets these requirements, actually exceeding the capabilities of many of the larger minis and mainframes originally used to implement LISP. Although most AI researchers have moved on to larger and faster machines, LISP on the IBM PC represents a significant and useful development tool.

LISP was originally designed as a development tool, but its flexible and extensible nature makes it quite efficient as a general-purpose language. In fact, good LISP compilers generate code that rivals the best FORTRAN and C compilers, even in numeric applications. LISP interpreters also compare well with BASIC in terms of speed. In complex implementations LISP has no real rivals.

Three LISP implementations are available for the IBM PC; all are respectable when compared to mainframe implementations. The first is IQ LISP, which runs under PC-DOS and is available from Integral Quality; the second is muLISP-82 by The Soft Warehouse, which runs under PC-DOS or CP/M-86 and is distributed by Microsoft Corp.; and the third is TLC LISP which runs under CP/M-86 and is available from The LISP Company (TLC).

BASIC LANGUAGE STRUCTURE

LISP is a simple language whose power derives from its extensibility and flexibility. It is usually easier to learn LISP as a first language than as a second or third language, because LISP's methods differ significantly from those usually employed by other languages. In any case, however, LISP can usually be picked up quickly.

The basic LISP objects are *lists* and *atoms*. *Lists* are sequential collections of objects—any objects. LISP normally prints lists surrounded by parentheses: (1 Big (LIST)). *Atoms* are usually symbols, numbers, strings or a host of other primitive objects depending upon the implementation. LISP uses these basic objects in evaluating and defining functions.

The evaluation of an atom returns the value of that atom; evaluation of a list consists of evaluating the first item in the list. The result of

LISP systems usually require large amounts of memory to operate effectively; the IBM PC, with an address space of 1 megabyte, meets these requirements, actually exceeding the capabilities of many of the larger minis and mainframes originally used to implement LISP.

that evaluation should be a function definition that is then applied to the remaining elements of the list. For example, (ADD 1 B) would apply the function definition for ADD to 1 and B. There are basically two types of functions in LISP: functions that evaluate their arguments and functions that do not. If we assume that ADD is one of the former, 1 and B would be evaluated and their results would be processed by ADD. The value of B should be a number, and ADD can return the sum of 1 and B since 1 evaluates to the number 1.

Evaluations can be nested to any level, since lists can contain other lists. If a list is a parameter, evaluating that parameter is the result of evaluating the list. Adding three numbers is as simple as (ADD 1 (ADD 2 3)). Discovering all the built-in functions in LISP can take a little time, but the syntax remains the same for each function. This is quite different from PASCAL or C, which have wonderfully complex syntax rules for various statements and precedence rules for expressions.

LISP does get a bit more complicated with regard to function definitions. A function is defined by evaluating that function; the resulting definition is then placed in a global definition list. Like many languages, LISP has different dialects, so check out any system you use before trying the following example:

```
(DEFINE (FACTORIAL X)
   (IF (EQUAL X 0) 1
        (TIMES X (FACTORIAL
  (DIFFERENCE X 1))) ) )
```

This is the typical recursive definition of factorial applied to some number x. If x is zero, the factorial is 1; otherwise,the result is x times the factorial of x minus 1. Note that the first element in each list in the definition refers to a function definition. In fact, FACTORIAL refers to itself. Even better, the application of FACTORIAL looks just like the applications of any of the built-in LISP functions, such as TIMES. One of LISP's nice properties is that it is easily extensible in a consistent fashion, since all function applications use the same format. This contrasts with languages like BASIC or PASCAL, which use a special syntax for FOR statements, another syntax for assignment statements, and so on. In fact, most other languages have special formats that can't be duplicated by a programmer.

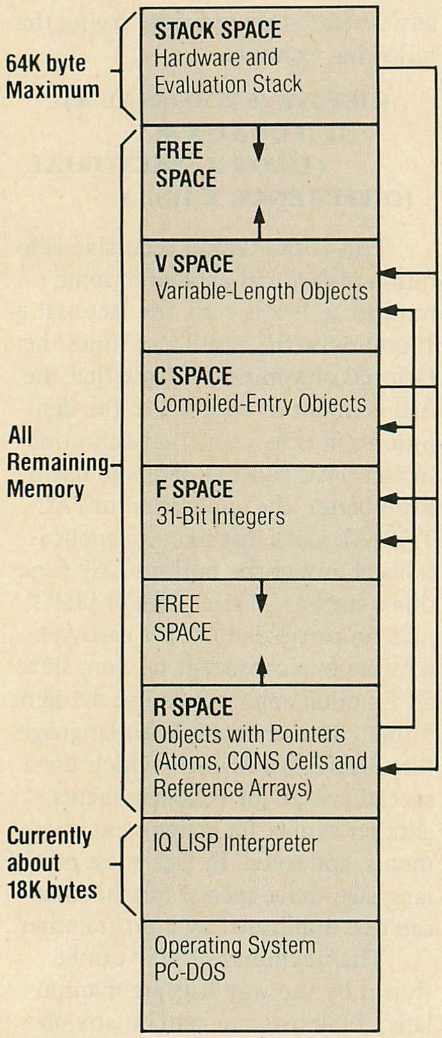
The flexibility of LISP can be shown by the way lists are manipulated. LISP lists can contain any object and can be any length. Lists can also be part of other lists, which

Bill Wong is on the technical staff of Rising Star, a software consulting firm. He is working on a Ph.D. in computer science at Rutgers University and is a frequent contributor to Microsystems.

Table 1:
Some Primitives in LISP

Expression	Result
A	(1 2 3 4)
(FIRST A)	1
(REST A)	(2 3 4)
(CONS 0 A)	(0 1 2 3 4)
(CONS A A)	((1 2 3 4) 1 2 3 4)

Figure 1:
IQ LISP Data Space. Note that all references to data-space areas come from the stack or R spaces.



helps the user establish efficient structures. This differs from conventional vectors or arrays, which normally contain the same type of elements and are fixed in length. Simple functions operate on any list, and more complex functions can be built using these primitives.

Some of the basic primitives are shown in table 1. The first expression is the variable *A*, which has a list as its value. This variable is used because each of the functions evaluates its arguments. In this case, evaluating *A* gives the list. The FIRST function is used to extract the first element of a list. The REST function gives the list minus the first element. Note that the list (2 3 4) returned by REST is actually the same portion of the list contained in *A*. CONS is shown twice to show how lists are CONStructed. In the first case, 0 is added to the head of the list. In the second case, the new list contains the list from A as its first element. Also, the REST of the list is also the list from A. It is not copied twice; new references to it have been created. These are simple features that are difficult to duplicate in most other languages.

LISP implementations tend to have many useful functions built in, including control functions like IF and LOOP, list-manipulation functions like APPEND and MEMBER, and a host of others. Obviously, there are a number of functions that must be known before any sophisticated programs can be written, but this is fairly easy once a familiarity with the syntax of the system has been acquired. In fact, with the information given above, it shouldn't be difficult to figure out the following definition:

```
(DEFINE (QUADRATIC X)
(SUM (TIMES A (SQUARE X))
    (TIMES B X)
       C))
```

One note about interacting with LISP. LISP normally has a PROPTREAD-EVAL-PRINT loop as its top-level interface to the user. This loop's definition is similar to this:

```
(DEFINE (LISP_TOP_LEVEL)
    (LOOP_FOREVER (PRINT
      PROMPT)
        (PRINT (EVAL
          (READ))))
```

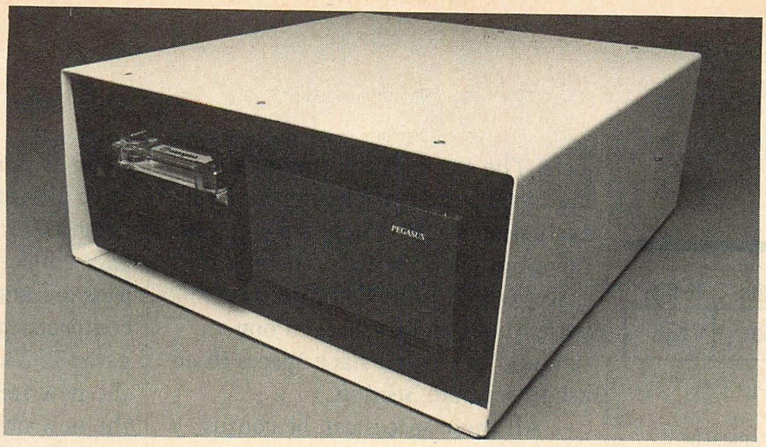
The system first PRINTs the LISP PROMPT and then READs an expression from the console. The expression is EVALuated, and the results are PRINTed. The loop continues forever or until the machine is turned off. Working with the system is simply a matter of entering LISP expressions. The immediate feedback, combined with other development tools such as a full screen editor, makes learning LISP an easy task.

LISP System Structures

Although LISP systems tend to be similar at the language level, they often differ radically at the system-structure level. LISP, like most languages, can be implemented in a number of ways, each of which has various advantages and disadvantages. The three LISP implementations discussed in this article—IQ LISP, muLISP-82, and TLC LISP—each use a different approach.

Optimizing Memory Capacity. IQ LISP uses a tagged architecture with a number of memory partitions in a large linear address structure, as shown in figure 1. There are two invariant parts: the operating system, which is not really part of IQ LISP, and the IQ LISP interpreter. All remaining space in the system is divided up into a number of spaces whose size can change during the operation of the LISP system.

The spaces or partitions are called R space, F space, C space, V space, stack space, and free space (actually two areas). R space contains objects with references (pointers) such as atoms, CON cells (used in LISP lists), and reference arrays. F space contains 31-bit fixed-point numbers, while C space contains an assembly language code. V space contains any variable-length objects that do not contain LISP references to other ob-



$1495 for a complete 10 megabyte hard disk system is a good deal — but not if you need 23 megabytes of storage!

That's why our 23 megabyte hard disk system is priced at only $2249. And for another $895, you'll get 23 megabytes of tape backup security.

The Pegasus 10 and 23 come complete with all the software and hardware you need to start operations.

If you've outgrown the storage capabilities of your IBM PC or compatible computer but haven't grown into the giant pricetags on 10- or 23-megabyte hard disk systems, you're in for a surprise.

The price on these complete systems featuring the latest components just reached an all time low!

And you get everything you need to start working on your hard disk system:

- Top quality formatted fixed hard disk (ST506 compatible)
- Top quality controller card
- Software that runs on DOS 1.1 and 2.0, CP/M 86
- Host adapter
- Integral power supply
- Cables
- External custom chassis with additional space for either our tape backup, another hard disk, or floppy disk drive.
- 90 day warranty

What's the catch?

There is none. It's a simple matter of economics. Pegasus saw the growing need for mass storage, made a commitment to fill this need, purchased thousands of the finest quality hard disks, and is now passing the good deal along to you.

The only thing you may be missing in buying the Pegasus instead of the IBM XT expansion chassis is something you may not need in the first place: eight expansion slots, a communications card, three little initials, and an extra $1,000 to $2,000 out of your pocket.

But don't I need a tape backup for all that storage?

We recommend it! That's why we're offering you the same great deal on a ¼" cartridge tape backup. We've watched other companies offer 10 and 20 megabytes of storage and forget about the backup altogether. Our tape backup, which carries a 90 day warranty, retails for just $895 — a small price for over 23 megabytes of formatted storage and lots of peace of mind.

Where can I buy a Pegasus hard disk system?

Pegasus systems are available only through dealers. So contact the dealer in your area who sells IBM PC or compatible computers. If he doesn't have Pegasus, chances are he'll carry it soon. Just ask him to call us. We will ship him your Pegasus unit immediately.

Does Pegasus have larger storage systems?

Absolutely. 65 and 140 megabyte systems will soon be available from Pegasus. And the best part is that they, too, are breaking new ground when it comes to pricing. If you've outgrown 10, or even 23 megabytes of storage, ask about the larger capacity units. All with the same top quality hardware and software and full 90 day warranty.

GREAT LAKES COMPUTER PERIPHERALS, INC.

2200 West Higgins Road, Suite 245
Hoffman Estates, Illinois 60195
In Illinois (312) 884-7272
800-323-6836

Dealer Inquiries Invited

CIRCLE NO. 122 ON READER SERVICE CARD

Figure 2:
IQ LISP Object Formats

2 Bits	1 Bit	1 Bit	4 Bits	16 Bits
Format	GC	Literal	Type	Value

Format	Description
0	Located in the stack space or subsequent item in a multiple-item element in R space
1	First item of an identifier element in R space
2	First item of an array element in R space
3	First item of a CONS cell in R space

GC	Garbage-collection marker bit

Literal	Description
0	Object is a reference, value contains the segment number, and type contains the offset
1	Object is a literal (see type)

Type	Description (valid only if literal field is 1)
0	Zero-length string
1	1-byte string in value field
2	2-byte string in value field
3	Small (16-bit) integer in value field
4	NIL, special LISP atom
5	(UNBOUND), special LISP value
6–15	Reserved

Value	See type and literal fields

jects. The stack space contains the hardware system stack and the LISP evaluation stack. (These are actually the same stack.) Finally, there is free space, which is divided into two parts, the R and F spaces in one section and the V and stack spaces in another, separate, section.

All these spaces can be confusing, but they each have a purpose. LISP supports a number of different object types, including atoms, lists, and integers. Objects can be dynamically allocated and de-referenced (not deallocated). When all references to an object are gone, deallocation is done automatically through a process called "garbage collection." Grouping similar types of LISP objects into spaces simplifies the allocation and garbage collection process.

The literal bit is used to determine whether the remaining portion of the block is a data object or a reference. Data objects are small literal values that can be encoded into the remaining type and value fields. References use the type and value fields to develop an 8086/8 address.

Why this particular grouping? The objects in the R, C, and stack spaces are the only ones that contain references to other spaces; these objects are thus the primary target for the garbage collector. In addition to referencing objects in their own space, objects can reference objects in other spaces, as is shown by the arrows in figure 1. Note that no reference is ever made to free space.

Free space is used to allocate new objects; those of a particular type are allocated from the edge of the nearest free space partition. The garbage collector is started when an object needs to be allocated and the free space partition has insufficient space. The garbage collector determines what objects are not referenced (garbage) and compacts each space by removing the garbage. The garbage then becomes the new free space area from which the new object is allocated. This collection and compaction process usually means that the spaces shift positions. The impact of such shifts will become more apparent after the object structure has been described.

Objects in IQ LISP can be divided into three general types: objects with references, fixed-size objects with no references, and variable-size objects with no references. Figure 2 shows the basic building block for objects with references. This block is a 3-byte entity that can refer to a literal object (itself) or to an object located elsewhere in the 8086/8 address space. The format of a block is determined by the two most significant bits, which determine the blocks' relationship within an object. These bits, along with the next bit, are used by the garbage collector during its collection process.

The literal bit is used to determine whether the remaining portion of the block is a data object or a reference. Data objects are small literal values that can be encoded into the remaining type and value fields. References use the type and value fields to develop an 8086/8 address. The 16-bit value field is used as the segment paragraph address allowing an object to reside anywhere in the 1-megabyte address space. The 4-bit type field is used as an offset into the paragraph. Getting the offset is simple: just AND the type field with 000F hex.

Figure 3 shows how the blocks can be concatenated into LISP objects. CONS cells are the basis for LISP lists and are usually quite numerous, so they are fairly small (6 bytes each). Identifiers, also called symbolic atoms, are larger (15 bytes each), but they tend to make up a smaller part of the LISP system. Reference arrays

Figure 3:
IQ LISP R-Space Objects

Format	GC/Literal/Type/Value Fields
3	CAR (first element)
0	CDR (second element)

CONS Cell (6 Bytes)

2	Value of identifier
0	Function definition
0	Property list
0	Print name
0	Next identifier in OBLIST

Identifier (15 bytes)

1	Size in bytes
0	Dimensions
0	Bound 0
0	
0	First element
0	

Reference Array
(12 bytes minimum 32K bytes maximum)
Elements in column order (ex. (0 0) (0 1) (0 2))

Figure 4:
IQ LISP V-Space Objects

1 Byte	2 Bytes	N Bytes
Type	Length (Optional)	Data

Type	Description	Length Field
0	Character string	Length of string
1	Byte array	Length of array
2	Positive long integer	Length in bytes
3	Negative long integer	Length in bytes
4	Small integer array	Length in bytes
5	Short floating-point array	Length in bytes
6	Long floating-point array	Length in bytes
7	File object	Length in bytes
8	Short floating-point number	none (5 bytes, implicit)
9	Long floating-point number	none (9 bytes, implicit)
10	Dead space	none (1 byte, implicit)
11–255	Reserved	—

are generalized extensions of CONS cells, but they are more efficient for large vectors. Note that all of these objects are a multiple of three bytes (one block); this structure leads to more uniform manipulation and eases the task of the garbage collector.

Objects in the F and C spaces have a fixed format and are less interesting than those in the R and free spaces. V-space objects are a bit more complicated, but not much (see figure 4). These objects are byte-oriented, with the first byte containing the object type, the next two bytes containing the length (if that length is not implied by the object's type), and the remaining bytes containing the data portion.

V-space objects are not collected in the same fashion as other objects because of two properties: first, V-space objects have no references to other objects and therefore cannot cause another object to be de-referenced, and second, each V-space object can be referenced by only one pointer. Therefore, a V-space object becomes garbage when the reference to it changes or becomes garbage. A comparison of the various structures in V space with the objects in the R space makes it clear that merging the two leads to a more complex job for the garbage collector.

Finally, there is the stack space, which can contain references and data of all sorts but is never garbage-collected. All space is allocated and deallocated using the standard stack process. Although the stack space is never garbage-collected, it does contain type markers on each stack frame, and the garbage collector does scan the stack for references to objects in the other spaces. The stack space contains the roots for all accessible objects that are located in the IQ LISP data space.

In summary, IQ LISP is a tagged LISP architecture with partitioned spaces and a compacting garbage collector. The unique 3-byte object format shown in figure 3 allows IQ LISP to support the entire 8086/8 address space while keeping object sizes to a minimum. This provides an inspired LISP implementation on a machine that does not lend itself easily to language implementation.

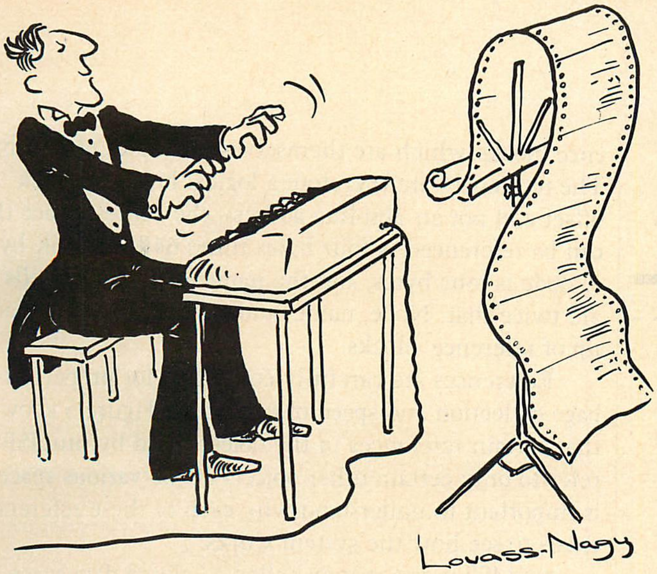
Optimizing Speed. muLISP-82 differs dramatically from IQ LISP. It offers faster execution speed, but at the cost of a possible reduction in system memory utilization. Whereas IQ LISP can access the entire 8086/8 address space using its 3-byte block references, muLISP-82 can utilize only 256k bytes more than are used by the interpreter and operating system (see

MuLISP-82 offers faster execution speed, but at the cost of a possible reduction in system memory utilization.

figure 5). However, many machines do not have more memory than muLISP-82 can use, so no space is wasted by using this system. To see why muLISP-82 has this limitation, we need to examine the structure of the objects in the muLISP-82 data space.

Some of these objects are shown in figure 5. The node, name cell, and number cell are objects made up of blocks that are similar to those in IQ LISP. However, these blocks are two instead of three bytes long and are not tagged. In place of tagging, muLISP uses what is called a typed partitioned data space with fixed-sized objects. muLISP determines an object's type by keeping a record of the location of each data-space boundary and comparing the object's address with the boundary addresses. The object's type is the same as that of the partition containing the object address. This identification system has the advantage of reducing the object-reference size by 50 percent.

How can a 16-bit pointer reference objects in a space of 256k bytes? The node, name, and number cells are built from a multiple of two refer-



•NEW PRODUCTS•

Before Johann Sebastian Bach developed a new method of tuning, you had to change instruments practically every time you wanted to change keys. Very difficult.

Before Avocet introduced its family of cross-assemblers, developing micro-processor software was much the same. You needed a separate development system for practically every type of processor. Very difficult and very expensive.

But with Avocet's cross-assemblers, a single computer can develop software for virtually any microprocessor! Does that put us in a league with Bach? You decide.

The Well-Tempered Cross-Assembler

Development Tools That Work

Avocet cross-assemblers are fast, reliable and user-proven in over 3 years of actual use. Ask NASA, IBM, XEROX or the hundreds of other organizations that use them. Every time you see a new microprocessor-based product, there's a good chance it was developed with Avocet cross-assemblers.

Avocet cross-assemblers are easy to use. They run on any computer with CP/M* and process assembly language for the most popular microprocessor families.

5¼" disk formats available at no extra cost include Osborne, Xerox, H-P, IBM PC, Kaypro, North Star, Zenith, Televideo, Otrona, DEC.

Turn Your Computer Into A Complete Development System

Of course, there's more. Avocet has the tools you need from start to finish to enter, assemble and test your software and finally cast it in EPROM:

Text Editor VEDIT -- full-screen text editor by CompuView. Makes source code entry a snap. Full-screen text editing, plus TECO-like macro facility for repetitive tasks. Pre-configured for over 40 terminals and personal computers as well as in user-configurable form.

CP/M-80 version $150
CP/M-86 or MDOS version $195
(when ordered with any Avocet product)

EPROM Programmer -- Model 7128 EPROM Programmer by GTek programs most EPROMS without the need for personality modules. Self-contained power supply ... accepts ASCII commands and data from any computer through RS 232 serial interface. Cross-assembler hex object files can be down-loaded directly. Commands include verify and read, as well as partial programming.

PROM types supported: 2508, 2758, 2516, 2716, 2532, 2732, 2732A, 27C32, MCM8766, 2564, 2764, 27C64, 27128, 8748, 8741, 8749, 8742, 8751, 8755, plus Seeq and Xicor EEPROMS.

Avocet Cross-assembler	Target Microprocessor	CP/M-80 Version	•CP/M-86 IBM PC, MSDOS** Versions•
•XASMZ80	Z-80		
•XASM85	8085		
XASM05	6805		
XASM09	6809		
XASM18	1802		$250.00 each
XASM48	8048/8041		
XASM51	8051	$200.00 each	
XASM65	6502		
XASM68	6800/01		
XASMZ8	Z8		
XASMF8	F8/3870		$300.00 each
XASM400	COP400		
XASM75	NEC 7500	$500.00	
Coming soon: XASM68K...68000			

(Upgrade kits will be available for new PROM types as they are introduced.)

Programmer $429
Options include:

- Software Driver Package --
- enhanced features, no installation
- required.
- CP/M-80 Version .... $ 75
- IBM PC Version .... $ 95
  RS 232 Cable .... $ 30
  8748 family socket adaptor . . . $ 98
  8751 family socket adaptor . . . $174
- 8755 family socket adaptor . . . $135

- **G7228 Programmer by GTek** -- baud
- to 2400 ... superfast, adaptive program-
- ming algorithms ... programs 2764 in one
- minute.

- Programmer $549

- Ask us about Gang and PAL programmers.

- **HEXTRAN Universal HEX File Con-**
- **verter** -- Converts to and from Intel,
- Motorola, MOS Technology, Mostek,
- RCA, Fairchild, Tektronix, Texas
- Instruments and Binary formats.

- Converter, each version $250

Call Us

If you're thinking about development systems, call us for some straight talk. If we don't have what you need, we'll help you find out who does. If you like, we'll even talk about Bach.

CALL TOLL FREE 1-800-448-8500
(In the U.S. except Alaska and Hawaii)

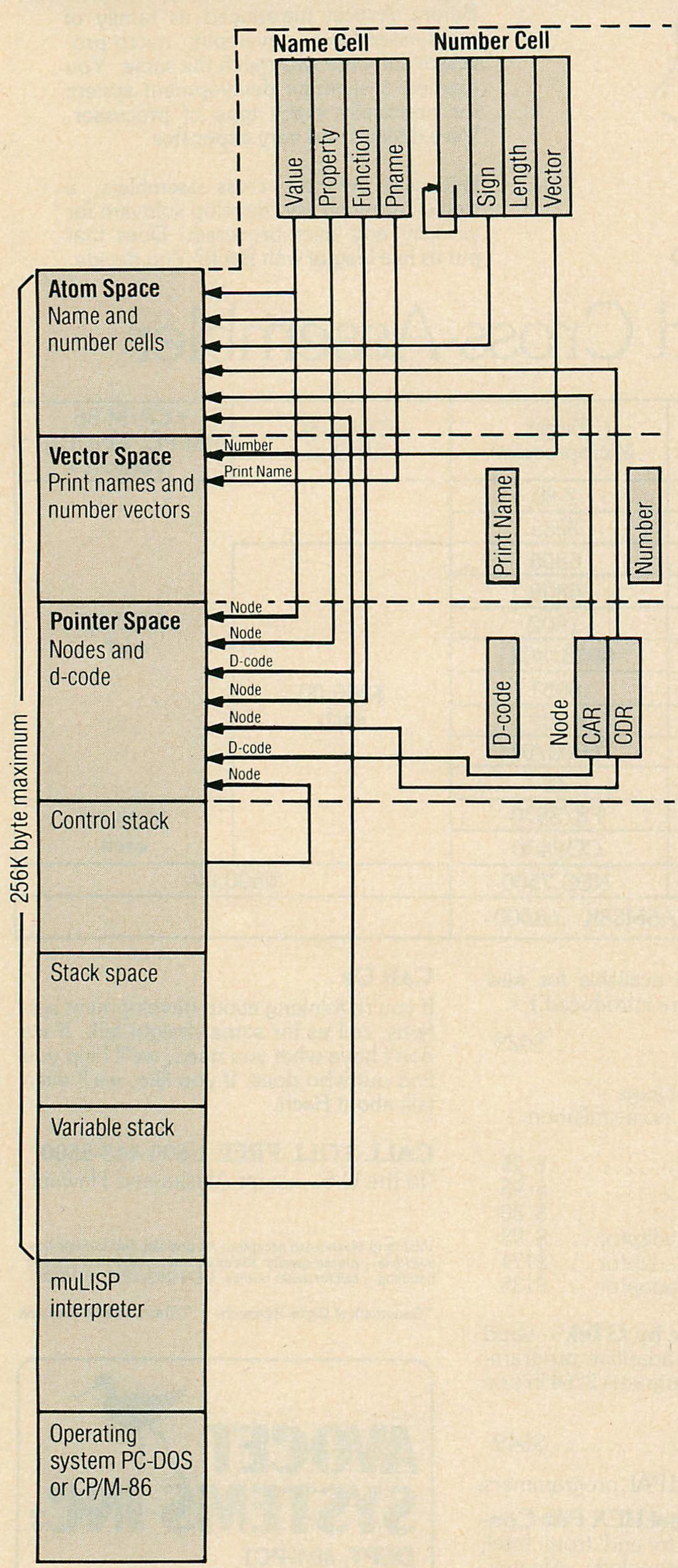
VISA and Mastercard accepted. All popular disc formats now available -- please specify. Prices do not include shipping and handling -- call for exact quotes. OEM INQUIRIES INVITED.

*Trademark of Digital Research **Trademark of Microsoft

AVOCET SYSTEMS INC.™
DEPT. 484-PCT
804 SOUTH STATE STREET
DOVER, DELAWARE 19901
302-734-0151 TELEX 467210

CIRCLE NO. 110 ON READER SERVICE CARD

Figure 5:
MuLISP "Closed" Data Space. All pointers reference items within the data space. All CAR and CDR pointers reference cells or nodes that include a CAR and CDR.



ence blocks, which are themselves multiples of four bytes. The references are therefore a logical building block offset and not an absolute address; the address space that can be referenced is four bytes times 64K, or 256K bytes. A node is four bytes, and the name and number cells are twice that. Node, name, and number cells are made up of reference blocks.

References are carefully controlled for proper garbage collection and speed of execution. Figure 5 shows that certain references in the objects used by muLISP can refer to only certain other objects in the various spaces. It is important to understand why each of these references exists to see how the system works.

muLISP uses a system called a "closed data space." It is similar to the IQ LISP system, in which all references are to objects within the LISP data space. muLISP places an additional requirement on the structure of the reference objects. Each object must be at least two reference blocks long and the first two reference blocks can refer only to objects with the same restrictions. These restrictions are imposed to increase the operation speed of the LISP system; accessing the first and second blocks (CAR and CDR) is the most common LISP operation, and increasing the speed of this operation would obviously increase that of the entire LISP system. In this case, the decrease in overhead is achieved by eliminating any type checking when accessing these fields; all subsequent references to objects using these fields will in turn reference an object of a similar structure. All references eventually lead to a circular reference list within the LISP data space. There is no way to get outside the space.

This method of operation has advantages and disadvantages. It is fast and performs uniform operations on all reference objects, but it hinders program development by permitting operations such as CAR and CDR to be done on any object, whatever that object's type. In IQ LISP this does not happen. For example, it is possible to take the CAR of a number cell in muLISP, but an error occurs if the same operation is tried in IQ LISP.

The third and fourth blocks are in a name or number cell. muLISP does type-checking when accessing these fields, because the references are to objects that have a special representation, not to those references discussed above. muLISP has "infinite precision" or "BIG" integers only for numbers that are similar to the character strings used for print names. The length of each object in muLISP, like that of objects in IQ LISP's V space, is variable. muLISP objects are garbage collected in the same fashion as the IQ LISP V-space objects.

A final type of object is the "distilled code" (d-code) object. It is not a p-code (pseudo-code), as in some Pascal implementations, but a linearized version of LISP optimized for execution and compactness. Although LISP structures are efficient, d-code can reduce the size of a LISP function by 20 to 35 percent. All user-defined muLISP

MULTI FUNCTION & RAM EXPANSION

AST RESEARCH

SIX PAK PLUS - Clock/Calender, 1 Serial Port, 1 Parallel Port (Expandable to 384K + Game Port) **$229**

MEGAPLUS II - Clock/Calender, 1 Serial Port, (Exandable to 512K, 2nd Serial Port, 1 Parallel, + 1 Game Port) Incl. Superdrive and Superspool **$229**

OPTIONS

64K Ram - set of 9 64K chips **$55**
Serial Port #2 .... **$40**
Parallel Port **$40**
Game Port **$40**
Expansion memory board for *Megaplus II* .... **CALL**

MAYNARD ELECTRONICS

Sandstar™ Series .... **CALL**

QUADRAM CORP.

QUADLINK - The IBM-APPLE Link Incl. System Software and On-Board Computer with 64K, Game Port, Display Generator, Disk Interface **$558**

PRINTERS

DOT MATRIX

EPSON	FX-80 (160cps)	**CALL**
	FX-100 (160cps)	**CALL**
OKIDATA	92 (120cps)	**$525**
	93 (120cps)	**$849**
NEC	8023 (100cps)	**$495**

DAISYWHEEL

BROTHER	HR-1 (16cps)	**$739**
NEC3500	Spinwriter (33 cps)	**CALL**
QUME	Sprint II Plus (40 cps)	**$1,475**

8" THINLINE FLEXIBLE DISK SUBSYSTEM

(Shown Below) Fully Assembled & Tested With Complete Documentation with Cable & Software

1-8480 - Dual Drive, Double-Sided, 2.4MB **$1,495**

1-8481 - Single Drive, Double-Sided, 1.2MB; Incl. Filler Panel - Second Drive Can Be Added Later **$995**

SOFTWARE UTILITIES FOR IBM® FROM TALL TREE SYSTEMS

J FORMAT-2, WINDRIVE, JETDRIVE, JSPOOL & JSORT .... **each $35**

DISK CONTROLLER FOR IBM® PC

MAYNARD ELECTRONICS

FDC 5¼"-Floppy Drive Controller **$189**
FDC/P8 8"-Floppy Controller **$195**

DRIVES FOR THE IBM® PC

5¼" FULL HEIGHT— Internal Mounting Flexible Disk Drives

TANDON

TM 101-4 .... **$265**

CONTROL DATA 9409 $225

Tandon TM 100-2 $205

Introducing

Tandon XTra™

NEW 10 MB WINCHESTER SUBSYSTEM

External plug-in and run includes case, power supply, cable-interface, complete documentation, and 1 year warranty.

Model TM-5112 LIST $1,995.00

NOW $1,595.00

Entire sub-system manufactured by TANDON CORP.

HERCULES

PCPI PRODUCTS

Qume. A Subsidiary of ITT

AMDEK

EPSON

MICROXPRESS

USI

MAYNARD

QUADRAM CORPORATION

UR UNIVERSAL RESEARCH

AST RESEARCH INC.

brother.

COLOR GRAPHICS & GRAPHICS CARDS

HERCULES	Graphics Card	**$380**
UNIVERSAL RESEARCH	Multidisplay Card Par Port	**CALL**
PLANTRONICS	Colorplus Par Port with the **DRAFTSMAN** Software	**CALL**

MONITORS

AMDEK

VIDEO 300
12" Hi-Res Green **$144**

VIDEO 310A
12" Amber Monitor .... **CALL**

COLOR I
13" Composite Color **$299**

COLOR II
Hi-Res RGB Monitor .... **CALL**

PGS

HX-12
12" Hi-Res Color (RGB) **$499**

1400C
12" Color Monitor **$325**

MODEMS

HAYES

SMART MODEM 300 .... **$230**
SMART MODEM 1200 .... **CALL**
SMART MODEM 1200B .... **CALL**
SMARTCOM II SOFTWARE **$80**
CABLE **$19**

DRIVES FOR THE IBM® PC OR PC 'JR'

5¼" HALF HEIGHT - Internal Mounting Flexible Disk Drives (Includes Strap Kits to Mount in Full Height Footprint) 6 MS Step Rate

TANDON

TM 55-2 .... **$225**
TM 55-4 .... **$245**

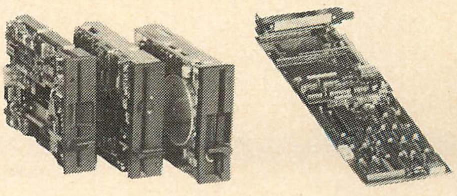
TEAC

FD 55-F **$249**

Tandon TM 50-2 $200

QumeTrak 142A $205

TEAC 55-B $215



(714) 632-8512

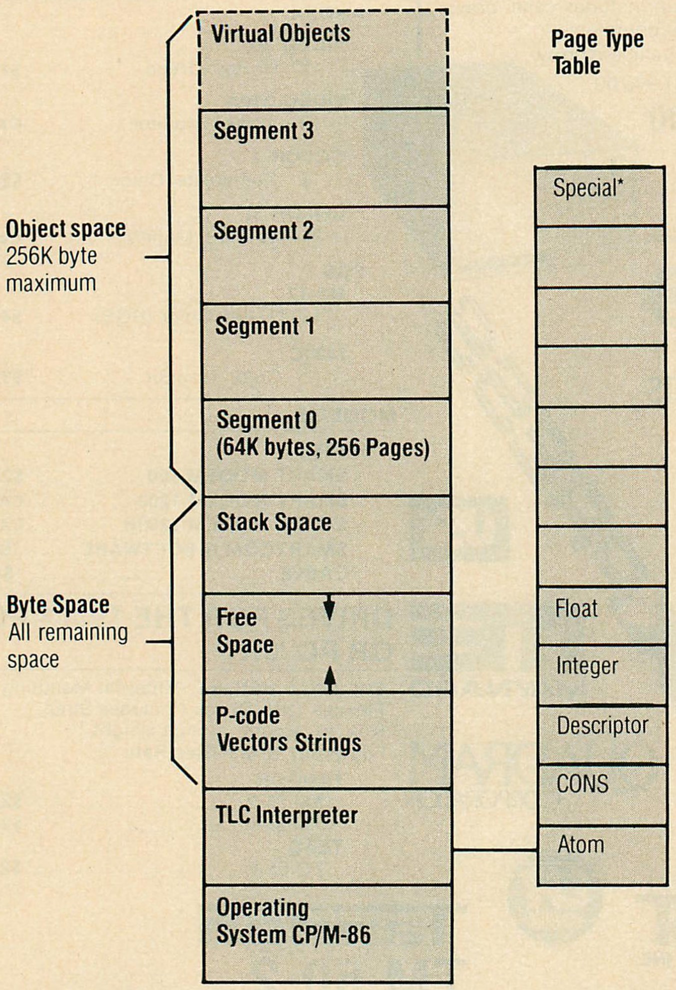
VISA MasterCard

MICROXPRESS
305 S. State College Blvd. Suite 135
Anaheim, CA 92806

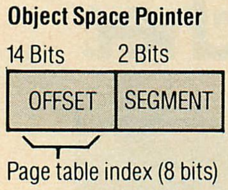
COMPETITIVE PRICES/PROFESSIONAL SERVICES
All equipment is brand new. We carry the full lines of the manufacturers included above. Call for products not listed. Combination and quantity discounts available on some products.

CIRCLE NO. 123 ON READER SERVICE CARD

Figure 6:
TLC LISP Data Space



BIBOP (BIg Bag Of Pages)
Object Allocation Method
Each page in the object space has a corresponding page-type able entry. All objects in a page are of the same type. Virtual objects can be referenced but do not use any space.



***Special Virtual Object Types**
NIL
Unbound
Single characters (0–255)
Small integers (±1024)

functions are automatically compiled into d-code. There is also a decompiler that gives the original LISP definition from d-code; this allows changes to function in an interactive mode. D-code is treated specially, and references to it occur only in the control stack or from a name cell.

Another interesting characteristic of muLISP is that there are three stacks (two are shown in figure 5). The control stack is used to keep track of the LISP function evaluation. References are made to d-code objects. The variable stack uses variables used in the function invocations to refer to other LISP objects, such as nodes, names, and numbers. Finally, there is the 8086/8 hardware stack, which is hidden in the muLISP interpreter.

As I mentioned earlier, stacks do not need garbage collecting to recover space, but they are used by the garbage collector to find garbage, which then becomes free space. (Free space is not shown in figure 5.) There are separate free-space pools for each object, such as name cells, number cells, print names, and so on. This is required since the type of each object is determined by its address with respect to a space boundary.

For efficiency and dynamic memory allocation, the garbage collector is a compactor, and space boundaries are moveable during compaction. This allows free space from one pool to be given to another pool as needed.

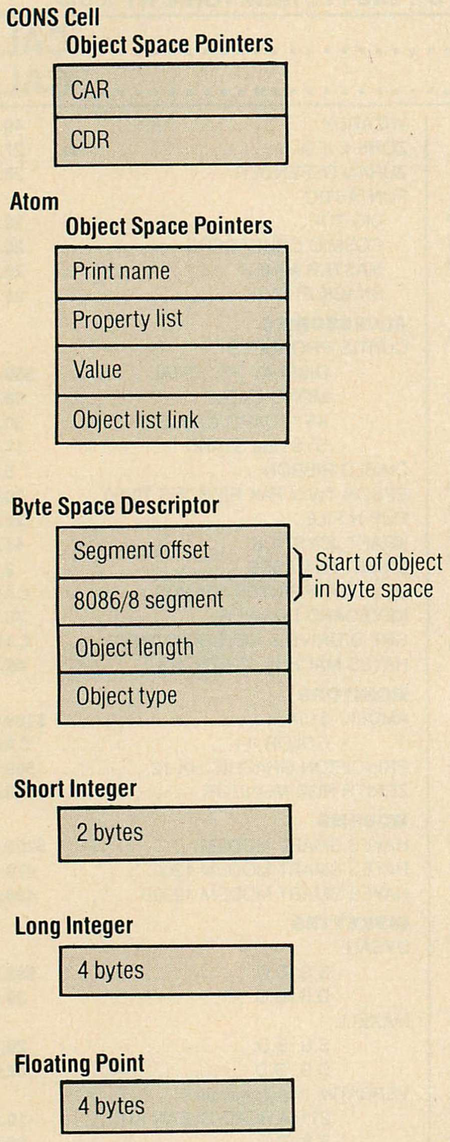
Compromising for Both Speed and Memory. TLC LISP uses an allocation method called BIBOP, for BIg Bag of Pages. BIBOP is a tagged architecture that is a blend of the IQ LISP tagged architecture and the muLISP partition architecture. Like muLISP, TLC LISP uses 16-bit references, but TLC LISP can utilize more memory than the muLISP implementation (see figure 6).

TLC LISP divides memory into two spaces (excluding the interpreter and operating system). The first is object space, which contains reference-type objects. The second is byte space, which is similar to the IQ LISP V space but may also contain LISP reference vectors that refer to items in object space or in byte space. Although this is confusing, it greatly expands the operation of the system.

Byte-space objects are simpler than object-space objects. There exists only one reference to an object in byte space; this reference is from either object space or a LISP reference array element in byte space. De-referencing an object in byte space causes it to become free space that can be reused immediately, unlike objects in object space, which must be recovered by garbage collection. Byte space contains objects like strings, vectors, and p-code. These objects tend to be of differing size and purpose. Free space is kept as a linked list of deallocated objects. Adjacent free space is merged with new free space into one larger space.

New objects are allocated in the TLC system by scanning the list of a portion of free space large enough to accommodate the new object. This approach differs from the muLISP and IQ LISP compacting garbage collector,

Figure 7:
TLC Object Space Objects



which keeps a free space pool as one large area at one end of a partition. The TLC approach can lead to a fragmented free-space pool, but it speeds deallocation significantly.

TLC LISP does have one large free-space pool, located between the stack and the allocated section of byte space, but this is simply viewed as a large free-space item at the end of the list. TLC LISP runs out of space if this free space is used up completely.

TLC's pointer-encoding scheme works well, since masking out the two least significant bits yields a segment offset that is a multiple of four; objects in object space have the same granularity.

Object space, which is more complex, uses the BIBOP scheme mentioned earlier. Because of system constraints similar to those in muLISP, object space is limited to 256k bytes. Like muLISP, TLC LISP uses 16-bit reference blocks and objects made up of multiples of 4 bytes. References are to 4-byte boundaries; hence the 256k-byte limit. TLC LISP can utilize more space than muLISP because byte space is separate and is not limited in size except by the capacity of the 8086/8 and the amount of memory that is available.

TLC LISP breaks up the reference pointer further by using the least-significant bits as a segment table index. There are four segment table entries with each segment being a 64k-byte 8086/8 hardware segment. The actual number and length of segments is determined at run time and depends upon the amount of memory available at that time. The segments need not be contiguous, although in CP/M-86, they can be; in some circumstances, contiguous segments may be desirable.

This pointer-encoding scheme works well, since the process of masking out the two least significant bits yields a segment offset that is a multiple of four; objects in object space have the same granularity.

Each segment is divided into 256 pages of 256 bytes. Each page has a corresponding entry in the page-type table, which is part of the TLC LISP interpreter. All objects on the same page are of the type specified by the entry in the page-type table. In a sense, this is similar to the muLISP boundary method of determining an object's type, but the TLC method is more efficient and faster due to the architecture of the 8086/8. The page-type table is usually a byte vector, and the table index is simply the eight most significant bits (most significant byte) of a reference. Accessing the most significant byte of a register in the 8086/8 is one instruction. Another is needed to index the table.

The maximum table size and the maximum number of pages is 1024, which is a BIg Bag Of Pages for a BIBOP scheme. Most pages are initially designated as free space, and pages are allocated as required with free objects within a page being reused before new pages of that type are allocated. However, some pages are reserved when the system is initialized and remain so throughout the life of the system. These pages are allocated in the page-type table, but they use no physical memory. They are called virtual objects in figure 6.

A virtual object is one that can be referenced and whose value is implied by its type and its position within a page. Since the reference implies the value, there is no need to reserve space for the object. The object space in a fully configured system thus really occupies less than 256k bytes. Virtual objects are those that occur frequently within a LISP environment such as NIL or small integers (−1024 to +1023). Fewer than 16 page-type table entries are used for these items. Computing a value from a reference is fairly easy even for numbers. For example, 128, 129, 130,

Three

Easy-to-Digest

Bytes of BASIC

If BASIC seems to be too much for you and your machine to master, The Brady Co. presents this universal language in three bytes! These three BASIC books are easy to swallow and easier to digest. Written to be readable and informative and substantially instructive, the three will take you through the language of BASIC from conversation to advanced applications.

The authors are experts who have taken great care to present the information you need, simply and logically. Buy one, buy two or all three and learn BASIC byte by byte!

"IBM PC*: An Introduction to the Operating System, BASIC Programming and Applications, Revised and Enlarged"

By Larry Joel Goldstein,
Martin Goldstein

To begin at the beginning, there is no better place to start than with the master teacher Dr. Larry Joel Goldstein. Written in an easy to read, self-study format, you'll find yourself communicating in BASIC in no time.

This updated and enlarged version of the classic introductory book starts with chapters called A First Look at Computers and Using Your PC for the First Time. It introduces you to BASIC and adds chapters on Controlling the Flow of Your Programming, Working with Data, Easing Programming Frustrations, Your Computer as File Cabinet and String Manipulation.

Chapters of special interest include an introduction to computer graphics and word processing. Begin BASIC with the work of the Goldsteins and you'll master it quickly!

1983/448 pp/paper/ISBN 0-89303-530-0/D5300-1/$18.95
Book/Diskette ISBN 0-89303-527-0/$39.95

"Handbook of BASIC for the IBM PC"

by David Schneider

This is the reference manual for the IBM PC* written especially for the beginner. It is organized by BASIC programming statements for easy use. The reader is able to go directly to the information they want when they want it.

The Handbook contains essentially the same information as the BASIC manual supplied with your IBM PC, rewritten and reformatted to make it easier to understand and apply. The book assumes that the reader is unfamiliar with BASIC and explains the manual to make it more valuable for everyday use.

1983/350pp/paper/ISBN 0-89303-506-8/D5068-4/$19.95
Book/Diskette 0-89303-508-4/$39.95

"Advanced BASIC and Beyond for the IBM PC"

by Larry Joel Goldstein

For those ready to go beyond basic BASIC, this is the advanced reference you've been seeking. It covers the advanced skills of BASIC programming including: focused coverage of files, graphics, even trapping, machine language and subroutines.

This guide offers an upper level, self-study course on BASIC which includes many ready-to-use programs. It also offers exercise sets with questions and answers supplied. It is a must for the IBM PC* programmer as the complete reference for information on BASIC instruction.

1983/400pp/paper/ISBN 0-89303-303-0/D3030-6/$19.95
Book/Diskette 0-89303 325-1/$44.95

Begin to advance in BASIC today with a book from The Brady Co., available at Computerlands, IBM Product Centers, Sears Computer Centers, Entré Computer Centers, B. Dalton Booksellers, Waldens, Crown Books, and other fine bookstores and computer dealers nationwide. Or call 800-638-0220 for information.

BRADY

The PC Specific Library

CIRCLE NO. 264 ON READER SERVICE CARD

The Brady Co., A Prentice-Hall Company, Bowie, MD. 20715

*IBM PC is a registered trademark of the International Business Machines Corporation.


INFINITE SOLUTIONS
Like the molecules in a snowflake, the elements of a computer database can be structured and related in an infinite variety of combinations. Being able to present these combinations quickly and efficiently, with maximum flexibility and minimum programming knowledge, is the mark of excellence which sets the sophisticated database management system apart from the ordinary.
Thousands of satisfied users, from amateur and professional programmers to government agencies, major corporations and industries, have found that DataFlex has no equal in applications development software.
Write or phone for our latest literature and a list of existing applications.
Compatible with CP/M, CP/M-86, MP/M-86, MSDOS, PCDOS, TurboDOS, Novell Sharenet, PC-Net, Molecular N-Star, Televideo MmmOST, Action DPC/OS, Omninet, IBM PC w/Corvus and OSM Muse.
DATAFLEX™
APPLICATIONS DEVELOPMENT SOFTWARE
DATA ACCESS CORPORATION
CIRCLE NO. 134 ON READER SERVICE CARD
8525 S.W. 129th Terrace, Miami, Florida 33156 (305) 238-0012 TELEX 469021 Data Access CI
MSDOS is a trademark of Microsoft. CP/M and MP/M are trademarks of Digital Research. DataFlex is a trademark of Data Access Corp.

and 131 would be the first four references of one of the virtual pages. The offset values for these references would be 0, 4, 8, and 12, since each

IQ LISP can utilize as much memory as the 8086/8 will allow; muLISP-82 sacrifices full memory utilization for speed and good utilization of the memory it does occupy; TLC LISP is close to IQ LISP in terms of memory utilization when large amounts of memory are available for use.

reference is on a four-byte boundary. Dividing the reference by four and adding the value of the first element gives the proper value for each reference. Because this approach requires only a few instructions to do the conversion, it saves space and time.

Figure 7 shows the form of the items in object space. The CONS cell is the basic LISP list-constructor and consists of two reference blocks. Atoms and descriptors use twice that amount of space, athough byte space descriptors contain different types of information. Numeric atoms such as integers and floating-point numbers are also basic object space types. This differs from the method used in IQ LISP, which places integers and floating-point numbers in V space.

TLC atoms have four reference blocks. The first is the print name, which usually refers to a byte space descriptor that itself defines a string containing the name of the atom. The second is the property list, which is followed by the value of the atom. TLC LISP differs from IQ LISP and muLISP in that the value of an atom is TLC LISP is also the function definition of the atom. IQ LISP and muLISP have separate references for the value and the function definition. The TLC method is more consistent when dealing with atoms in closures and lambda bindings. It also tends to deter people from defining a function that uses the same name as a variable, thereby making the use of an atom context dependent.

The TLC byte-space descriptor object has a structure different from that of the TLC atoms. The first four bytes are an 8086/8 segment/offset pair that can be loaded into an 8086/8 register pair with one instruction. The object type and 8086/8 address could probably have been encoded into 3 bytes as in IQ LISP. However, the resulting object would not have been a multiple of 4 bytes, which would make allocation difficult and would increase the amount of time required to generate a real address. The object length is a 16-bit signed integer that limits objects to 32k bytes—not a major restriction. The object type specifies one of many types, such as character strings, p-code vectors, integer vectors, reference vectors, and so on. As I mentioned earlier, only one byte-space descriptor will ever reference a byte-space object. Also, byte-space objects are stored contiguously.

TLC and IQ LISP store typical LISP function definitions as LISP lists, and muLISP stores them as d-code. TLC LISP also has the ability to evaluate functions defined as p-code (pseudo-code). This code is typically less than half the size of list-style definitions and runs two to three times faster. Although not as fast as machine code, p-code does take up less space. It offers a good blend of speed, compactness, and flexibility.

System Comparison

Each approach to implementing LISP discussed here has its advantages and disadvantages. IQ LISP can utilize as much memory as the 8086/8 will allow. Its memory allocation scheme is

reasonably compact and easy to use, but its execution time is slightly slower than that of the other two systems. muLISP-82 sacrifices full memory utilization for speed and good utilization of the memory it does occupy. The 50 percent reduction in space used for most of the LISP objects makes this LISP system ideal for systems with less than 500k bytes. TLC LISP comes between IQ LISP and muLISP in terms of speed when list LISP code is used; it is faster than either system when p-code is used. TLC

The IQ LISP manual is a good example of the improvements in documentation being made in the software industry, much to the relief of the users.

LISP is close to IQ LISP in terms of memory utilization when large amounts of memory are available. It, too, uses 50 percent less space than IQ LISP does for common LISP objects, but it can utilize more than 256k bytes, since many LISP objects can be in byte space. Ideally, muLISP and TLC LISP can fit 64K CONS cells in 256k bytes. IQ LISP would use 384 kbytes, but it could also support 170K CONS cells with a full megabyte, almost three times the other two.

All three systems provide respectable LISP systems; there are no "order of magnitude" differences among them. In the long run, the choice of a system will probably depend more on the higher-level functions and support provided with each system.

DOCUMENTATION

The IQ LISP manual is a good example of the improvements in documentation that are slowly being made in the software industry, much to the relief of the user community. The manual is contained in a sturdy three-ring binder, and its contents are as solidly designed as the binder. Although a determined programmer could use the system after thoroughly looking over this manual and its included support files, this is primarily a reference manual rather than an introduction to LISP. The manual does, however, refer the user to a number of good books on LISP.

The documentation is divided into large sections covering the basic system, supplied primitives, the development system, and finally implementation details. Indices and table of contents are standard and easy to use. Examples are short but sufficient. It is easy to read the information and verify the examples by ac-

tually using the system. The documentation even addresses timing issues, such as the time it takes to divide a 77000-digit number by a 38000-digit number. Although some details may seem esoteric, it is nice that this information is supplied.

The description of the system is complete and enlightening, especially for users interested in adding assembly language routines or seeing what can be done with the Intel 8086 processor. The details are presented precisely with excellent, although not too numerous, examples. All these factors help make the IQ LISP documentation excellent.

The muLISP/muStar documentation is a rehash of the 8-bit version of the same system, which is compatible. A number of changes and corrections have been made, but the flavor of the original manual remains. Unfortunately, although the physical construction of the manual and its binder is professional and durable, the content are less impressive. The manual is complete and reasonably well organized, having a table of contents and a couple of indices. Its main drawbacks are the lack of examples and the use of Backus-Naur Form (BNF) syntax to describe built-in LISP functions. Although BNF is concise, it leads to a confusing description of the various functions. The lack of examples adds to the confusion.

All aspects of the system are described in a terse but reasonably complete fashion. Use of an external text is required for anyone unfamiliar with LISP. Hopefully, the next version of the manual will be improved.

The TLC documentation is also an upgraded version of the 8-bit TLC LISP manual. A new version is expected to be published soon and will probably contain some improvements. As it is, the existing documentation is quite good. There are sometimes more examples than are necessary, but that is better than too few.

The manual is divided into four parts: a general discussion of LISP, a description of how TLC LISP works,

examples, and a description of all the built-in functions. The general discussion is not intended as a LISP tutorial but it is a good introduction. This manual, like the others, requires the support of one of the many books on LISP. The description of TLC LISP and the examples are nice but more detailed explanations would help. The descriptions of the built-in functions are grouped by functionality, which is good for learning but less desirable for general use; an alphabetical list or organization is needed for that. Overall, the quality of this manual falls between that of IQ LISP's and muLISP's documentation.

LISP OBJECTS

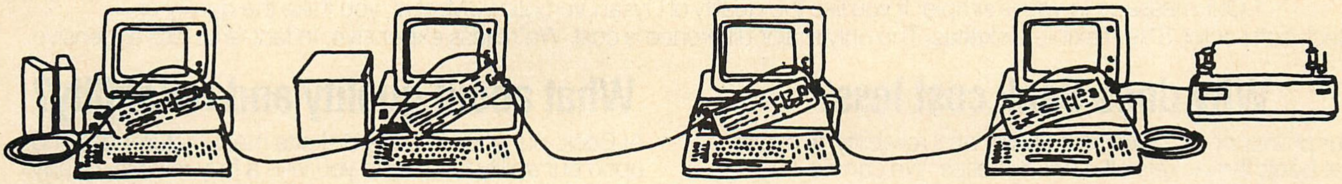
IQ LISP has a plethora of object types. It supports three kinds of integers (2-byte, 4-byte, and infinite-precision) and two kinds of floating-point numbers (4- and 8-byte IEEE compatible). Infinite-precision integers, although not really infinite, can be very large. They are actually variable-length byte strings. The length of the string increases with the magnitude of the number. Obviously, large numbers take more time to process than small numbers. There are many problems that require this type of precision, for which floating-point numbers are simply inadequate. The nice thing about using these integers is that programs with them hardly ever cause an overflow error, although they sometimes may run out of memory.

IQ LISP also supports variable-length character strings, the normal LISP list, and generalized arrays with arbitrary dimensions. Unfortunately, the arrays are restricted to 32K bytes in overall size, so increasing the number of dimensions does not help increase the overall capacity of an array. On the whole, IQ LISP contains almost all the support found on mainframe LISP implementations and greatly exceeds anything found in conventional languages such as FORTRAN and BASIC.

muLISP is a bit deficient in the area of object types, but it does have

PCnet™ and the B.O.S.S.™

Business-Oriented Software System for IBM-PC Networks



Multi-User Data Base Management Software Designed for Use by Non-Programmers

- **Easy to Use**
  First-time users often define multiple data bases, enter records, and produce reports within three hours. The B.O.S.S.™ is menu driven. It presents options and responds to commands in plain English, while preventing multi-user conflict without user programming.

- **Multi-User Simultaneous Updating**
  Several users can update the same database without worrying about losing data. The B.O.S.S. automatically prevents multi-user conflicts. Sophisticated record level locking protection procedures are used.

- **Handles Large Files**
  The B.O.S.S. handles business oriented data files with thousands of records as well as it handles small files. Sorting rarely is needed — The B.O.S.S. updates all indexes (up to 15) when changes are made.

- **Generates Reports Fast**
  The B.O.S.S.'s full-featured Report Program Generator produces sophisticated business reports fast. First time users can select menu options to design inquiries that match records in multiple files and perform computations.

- **Versatile, Powerful and Portable**
  The B.O.S.S. is fully capable on most 16-bit microcomputers and operating systems, including MSDOS™, CP/M-86™, and MP/M-86™. Applications designed on one system may be transported easily to another. The B.O.S.S.'s full power and versatility are most apparent when used to create application systems for 16-bit machines such as the IBM-PC™ and to create applications for networking systems such as PCNet™ (Allows addressing of 1 full megabyte of RAM).

- **How It Works...**
  Non programmers use Levels I and II.
  Level I consists of:
  - **A foundation** module used to define individual data files, enter data into those files, and to display, edit and print the data.
  - A powerful, fast **sort** capability.

  Level II consists of:
  - An **inquiry** function used to design and produce video pages.
  - A **report generator** that performs sophisticated calculations and produces user designed output to the display screen, printer or other files. The inquiry and report modules can access up to 8 separate data files simultaneously.

  Level III — Database modules for programmer use.

COMPARISON OF POPULAR DATA BASE MANAGEMENT PROGRAMS

	The B.O.S.S.	dBase™ II	TIM™ III	Condor™ 20	Data-Star™
Multi-user network	Yes	No	No	No	No
Menu driven	Yes	No	Yes	Yes	Yes
Files open on line	8	2	N/A	2	1
Maximum records/file	100,000,000	65,600	32,800	32,800	32,800
Maximum bytes/record	10,000	1,000	2,400	1,000	255
User must be programmer	No	Yes	No	No	No
Maximum number of indexes automatically updated.	15	1	1	1	1

(Data are taken from most recent versions of programs available to American Planning Corporation, Alexandria, VA)

- **Advanced Features**
  Experienced analysts and programmers use all levels to create sophisticated business application programs.
  The B.O.S.S. provides analysts and programmers with a broad and varied set of functions patterned after mainframe software development tools. Large and complex programs are created easily and in record time.

- **Savings**
  When used to develop programs, The B.O.S.S. reduces costs by at least 60% and slashes required time by up to 95%.

- **30-Day Money Back Guarantee**
  The B.O.S.S. package includes a 30-day, no questions-asked money back guarantee. Unlimited backup copies are allowed and the complete source code is provided so that advanced programmers can further tailor the program. Updates will be furnished for a nominal fee.

B.O.S.S. is a trademark of American Planning Corporation
PCNet is a trademark of Orchid Technology
CP/M & MP/M are trademarks of Digital Research, Inc.
IBM PC is a trademark of IBM Corporation
TIM III is a trademark of Innovative Software
dBase II is a trademark of Ashton-Tate
Condor is a trademark of Condor Computer Corporation
DataStar is a trademark of MicroPro, Inc.

CIRCLE NO. 139 ON READER SERVICE CARD

AMERICAN PLANNING CORPORATION

4600 Duke Street, Suite 425
Alexandria, VA 22304

1-800-368-2248
(In Virginia, 1-703-751-2574)

Dealer Inquiries Invited

Visa and Master Charge Accepted

Quality you expect, at a price you don't.

BECK DOUBLE DENSITY DISKETTES

SINGLE SIDED **$2.19 ea.** / **$2.79 ea.** DOUBLE SIDED

Our message to you is simple. If you like the quality of Dysan, Verbatim, 3M, et al, you'll like the quality of Beck soft sector, 5¼″ flexible diskettes. The only major difference is cost. We're less expensive. In fact, a lot less expensive.

Why does Beck cost less?

Our philosophy is: Excellent quality and reliability, at a cost that beats the jackets off other diskettes. We can do it because we (1) put our money into the product, not mega-marketing schemes and fancy packaging; and (2) sell our money-saving 25-diskette pack to you direct via a toll free order line, so you get fast, door-to-door service efficiently.

When you buy Beck, you've got the best.
Beck Quality. Beck Reliability.
And, of course, Beck Price.

1D, soft sector 5¼″ diskette $2.19 each
2D, soft sector 5¼″ diskette $2.79 each

For IBM, Apple, TRS and 97% of popular microcomputers.

What about quality and reliability?

At Beck, our success as a diskette manufacturer depends upon our ability to provide you with a fully reliable, quality diskette – every time. For that reason we take no shortcuts. You get the best because we are committed to excellence. Every diskette is manufactured to very strict quality standards. We test and retest 21 times throughout the manufacturing process to insure compliance with no less than 42 rigid specifications. We make sure you get the very best – a 100% certified, 100% error free diskette.

Our satisfaction money-back guarantee and full 7 year warranty are proof of our commitment to excellence and confidence in our product.

(in New Hampshire call 924-3821) ***Door to Door in 48 hrs.***

Order Now Toll Free

VISA MasterCard COD'S CASH ONLY

Corporate Accounts Welcome

1-800-BECK MFG

Order Toll Free 1-800-232-5634. Available in 25 pack only, plus freight, Complete with hub reinforcing rings, Tyvek envelopes, color coded user labels, and nonmetallic write protect tabs. All Beck Diskettes meet or exceed ANSI specifications.

CIRCLE NO. 193 ON READER SERVICE CARD

Books on LISP

Anatomy of LISP, by John Allen; McGraw-Hill; New York; 1978

This is an excellent book by one of the authors of TLC LISP. It describes the design philosophy of LISP and a LISP system including machine interface and structures. Allen addresses many of the problems and solutions encountered in building a LISP system and also covers LISP program design. Although not a text for novice LISP users, this book does provide interesting reading for persons wishing to understand the operation and reasoning behind a LISP implementation.

Thinking about [TLC] Logo, by John Allen, Michael Burke, and John F. Johnson; Holt, Rinehart, and Winston; NY; 1983

Logo is a fairly new language derived from LISP and currently in vogue in the educational field. Logo and LISP are so similar that the ideas presented in this book can be applied directly to a LISP system. In fact, the turtle graphic system described matches the TLC LISP implementation. The only difference between the two languages is in syntax, and that difference is only in the placement of parentheses (LISP) and of line numbers (Logo). The book is highly entertaining. In addition to learning about turtles (those little drawing devils) and ducks, the reader will encounter all sorts of programming techniques and ideas presented in a fresh way.

Artificial Intelligence Programming, by E. Charniak, Riesbeck, and McDermott; Lawrence Erlbaum Associates; Hillside, NJ; 1980

This book is a good source of LISP programming techniques and examples for the more advanced LISP programmer. It is referenced in the documentation of the various PC LISP implementations, and most of the examples can be used exactly as they appear. The book also contains some artificial intelligence methods and ideas.

Let's Talk LISP, by Laurent Siklossy; Prentice-Hall; Englewood Cliffs, NJ; 1976

An excellent source of information on the internal operation of the theory of LISP, this book also contains a good description of LISP. The examples are intended to show the operation, not the application, of the language. Required reading for anyone interested in the workings of LISP.

INTERLISP Reference Manual, by Warren Teitelman; Xerox Corp., Palo Alto Research Center, Palo Alto, CA; 1978

This giant volume, which describes one of the most sophisticated LISP implementations on a mainframe, is of interest here because many of the ideas described in this manual are used in the more advanced LISP systems, including those on the PC. This book is for those who understand LISP and want to learn more about various extensions that have been made, including coroutines, multitasking, and others; it is not recommended for the casual LISP user.

LISP 1.5 Primer, by C. Weissman; Dickenson Publishing Co.; Belmont, CA; 1967

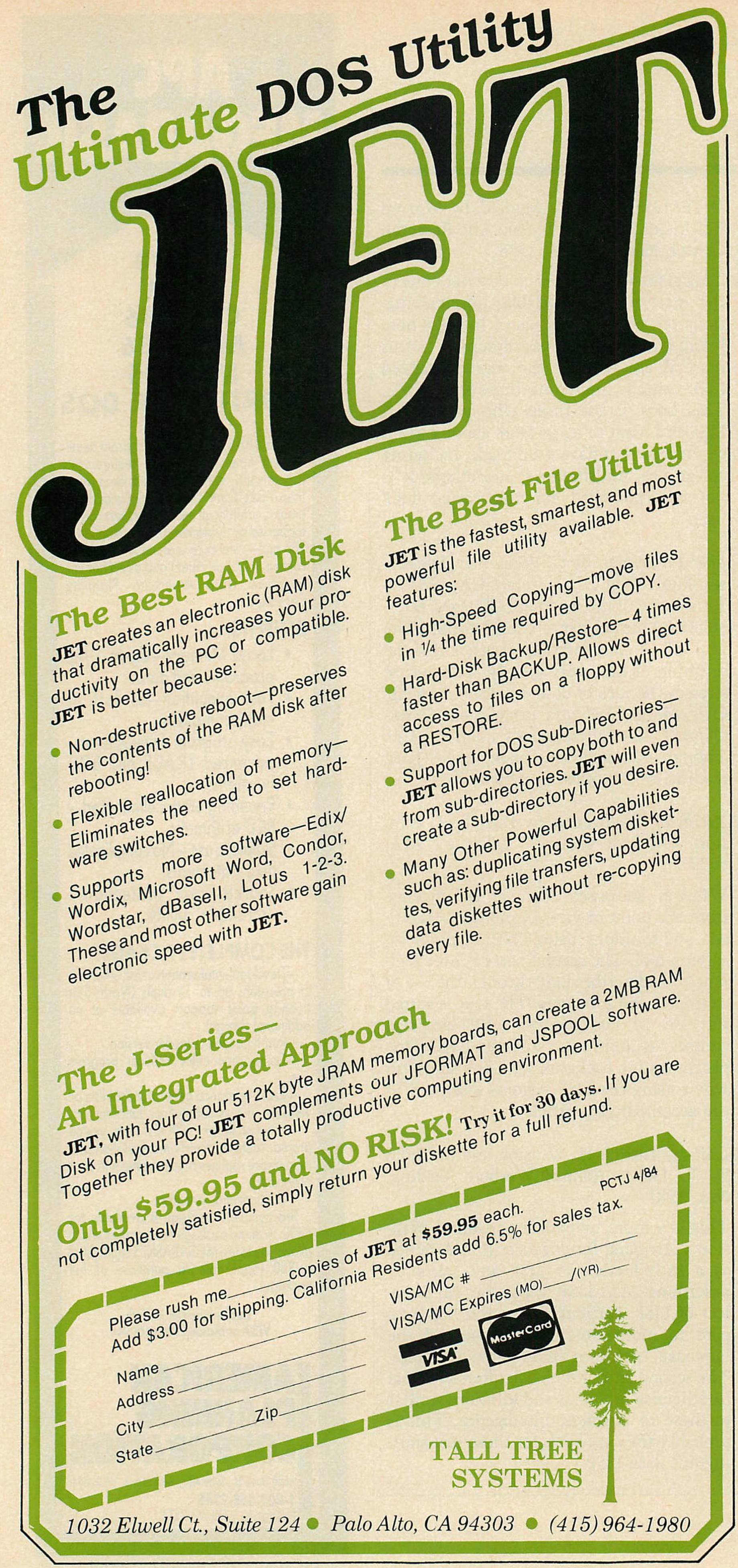
This is one of the original LISP texts, referenced by almost every book on LISP. The muLISP implementation matches this description of LISP very closely. The author stresses the idea of dotted pairs and EVALQUOTE for LISP/user interaction. EVALQUOTE has fallen out of common use. Though outdated, this book makes good reading for historical purposes.

Artificial Intelligence, by Patrick H. Winston; Addison-Wesley; Reading, MA; 1979

Winston addresses artificial intelligence from the point of view of problems, methods, and solutions, using LISP as a problem-solving tool. The second part of this excellent book describes LISP through the use of examples and problems. The text is suited for use in college-level LISP language courses, assuming the students have some programming background. It is also useful when using the LISP systems discussed in the accompanying article.

LISP, by Patrick H. Winston and Berthold K. P. Horn; Addison-Wesley; Reading, MA; 1981

This is a more up-to-date version of Winston's *Artificial Intelligence*. It is designed as a introduction to LISP and starts by describing MACLISP, a popular MIT dialect of LISP. Particular attention has been paid to incremental presentation of the language, including step-by-step examples. The second part of the book offers artificial intelligence problems and solutions similar to those in *Artificial Intelligence*. This is probably the best book for those just starting out with LISP.



infinite-precision integers and 32k-byte character strings. Unfortunately, the character strings are in the form of atom names, and the string manipulation functions are not numerous, although there are enough of them for most purposes. muLISP does not support floating-point numbers or arrays of any kind. The system was designed primarily to support the muMath package, which uses only integers; hence its simple structure. This simplicity can be an advantage, however, since it is fairly easy to comprehend the entire system. Also, muLISP's lack of support for these additional types of objects makes it possible for the interpreter to be smaller than those for TLC LISP and IQ LISP.

TLC LISP ranks with IQ LISP in having a larger mixture of object types. Various integer sizes are supported, and infinite-precision numbers are promised for the next release. Only one type of floating-point number is supported at this time, but that is sufficient for many numeric applications. Single-dimension arrays (vectors) are the only type of array that is supported, but it is possible to provide arrays by defining LISP functions that work on vectors of vectors. The advantage to this is that the basic system doesn't have to provide the support, and asymmetrical arrays can be created. Vectors are limited to 32k bytes, but vectors of vectors aren't.

> *TLC LISP ranks with IQ LISP in having a larger mixture of object types.*

TLC LISP also supports a "class" system similar to that found in the Smalltalk language. In a class system, messages are passed to objects that perform operations to process the message. The actual methods used to process the message are hidden from the sender of the message. For more detail, read one of the several available good books on the subject.

I/O Support and Graphics

LISP systems of the past tended to provide a less-than-adequate I/O subsystem. IQ LISP is improved to make the system usable for file I/O and good for console support, including basic graphic support. File I/O is at the character level, and sequential files are very important. Random-file I/O is not yet supported, which precludes any serious file data-base applications. The most attractive feature of this I/O system is the multiple-window support with graphics. Windows can be dynamically defined and manipulated. The graphics are a bit crude, with only point and line functions, but the system is reliable. Windowing gives the basis for a good debugging and development system.

IQ LISP also provides good support for the functions keys on the IBM PC. It can trap keys and execute LISP functions as they are pressed. A stacking mechanism is available for saving and restoring the key to function environment. This process is very useful in general and makes interactive game development a snap.

muLISP uses an archaic I/O redirection with only one set of sequential files, which default to the console. It is sufficient for loading and saving LISP functions and objects, but not for much else. Cursor control on the terminal is about all there is. Even so, muLISP is able to support a screen editor, which is supplied as part of the system.

The most attractive feature of this I/O system is the multiple-window support with graphics. Windowing provides the basis for an excellent debugging and development system.

TC LISP is the best of the three systems in terms of I/O. Although it does not have built-in formatting capabilities like FORTRAN's or COBOL's, they can be added. TLC, like IQ LISP, does allow multiple files, but TLC LISP also supports random file I/O, making data-base applications viable. I/O support through the class system is also provided, which makes the development environment very flexible. Stream support from multiple sources is possible. This allows a programmer to merge various input sources, possibly from a mixture of files, devices, and functions. If it is not already a built-in function, it can probably be built in TLC LISP.

Window support at this time is very rudimentary. Unlike IQ LISP, TLC LISP has no separate scrollable regions, but the window support is otherwise similar. Graphics is currently provided through Logo-like "turtles" that have line-drawing capa-

bilities. This is equivalent to IQ LISP's graphics, although TLC LISP's are somewhat easier to use.

Resident Editors and Package Support

IQ LISP comes with a resident line-oriented structure editor written in LISP. It can be modified and extended, and multiple copies can run in various windows on the screen if you create the windows yourself. A line-oriented structure editor works something like this: a logical cursor can be moved left, right, up, and down through the elements of a LISP list; and the current list can be printed and modified. It is similar to using a line-oriented text editor, but

> *TLC LISP also comes with a very good screen editor, which is written in LISP. The LISP list definition or the faster p-code version can be run. The editor has a single menu line at the top and uses the arrow and function keys.*

it is necessary to know the way a LISP builds lists. It is not bad once you get used to it but still is no match for a screen editor.

IQ LISP has "package" support but no separate "name" spaces. The latter differ from the spaces described in the IQ LISP system structure. Packages are kept as files containing LISP definitions in the operating system. However, the definitions are all mixed together and then loaded into the LISP system. The IQ LISP package support provides a way of grouping the definitions together so all updates can be reflected in the file. IQ LISP can also collect together many other kinds of information, such as functions to be executed when a package is loaded or removed.

TLC LISP supports separate "name" spaces in its package support. Each name space is independent of the other logical spaces. For example, the value of the atom TEST in one name space is not altered if the value of TEST in a different name space is changed. This separation becomes important when trying to combine packages that use the same atom names. Each name space has a name assigned to it, and references to atoms can be made by prefixing the atom name with the space name.

Each package has a set of definitions that it imports and exports. Those that are exported can be referenced by other packages. If two packages using the same name are loaded in IQ LISP, the definition from the first package would be lost. With TLC LISP, however, the packages can be loaded into different name spaces, preserving both definitions and allowing each package to access the proper definition. TLC LISP still provides the same kind of update mechanism for packages that IQ LISP does.

TLC LISP also comes with a good screen editor, written in LISP. The LISP list definition or the faster p-code version can be run. The editor has a single menu line at the top and uses the arrow and function keys. Control codes are similar to a well-known word processor. The screen editor makes TLC LISP easy to use.

muLISP also has a screen editor that is, to a certain extent, menu driven. No menu line is displayed when the item being edited is shown. Even so, it makes such a difference to be able to edit while in LISP that the screen editor is a definite plus for muLISP. Unfortunately, muLISP does not have any type of package support.

Error-Handling and Development Support

Error-handling is one of the few primitives that must be a built-in option in LISP. It is as basic as CONS, CAR, and CDR. The language cannot be extended unless the proper primitives exist. On the other hand, devel-

opment support, including TRACE and BREAK support, can usually be added without any special primitives. Here LISP differs from conventional languages like BASIC and PASCAL.

IQ LISP supports the more sophisticated CATCH and THROW primitives along with a very good BREAK and TRACE package that can be combined with the window system to provide an excellent development system. All errors other than fatal system errors can be trapped and processed under program control or interactively. The LISP evaluation stack can be examined and modified.

IQ LISP includes a development subsystem that enhances the normal error routines with a multiple-window debug system. When an error occurs, the screen is split into two windows, a large one above and a small one below. The top window displays the error, the LISP expression that caused the error, the current state, and a menu of options. The lower window can be used to evaluate any LISP expression, including those that examine the current state of the system. The subsystem can be loaded as needed and need not be part of any final program developed using it. This development package is one of the best on the market for any language.

With muLISP, if the program runs, it will run quickly. If it doesn't run, good luck debugging it.

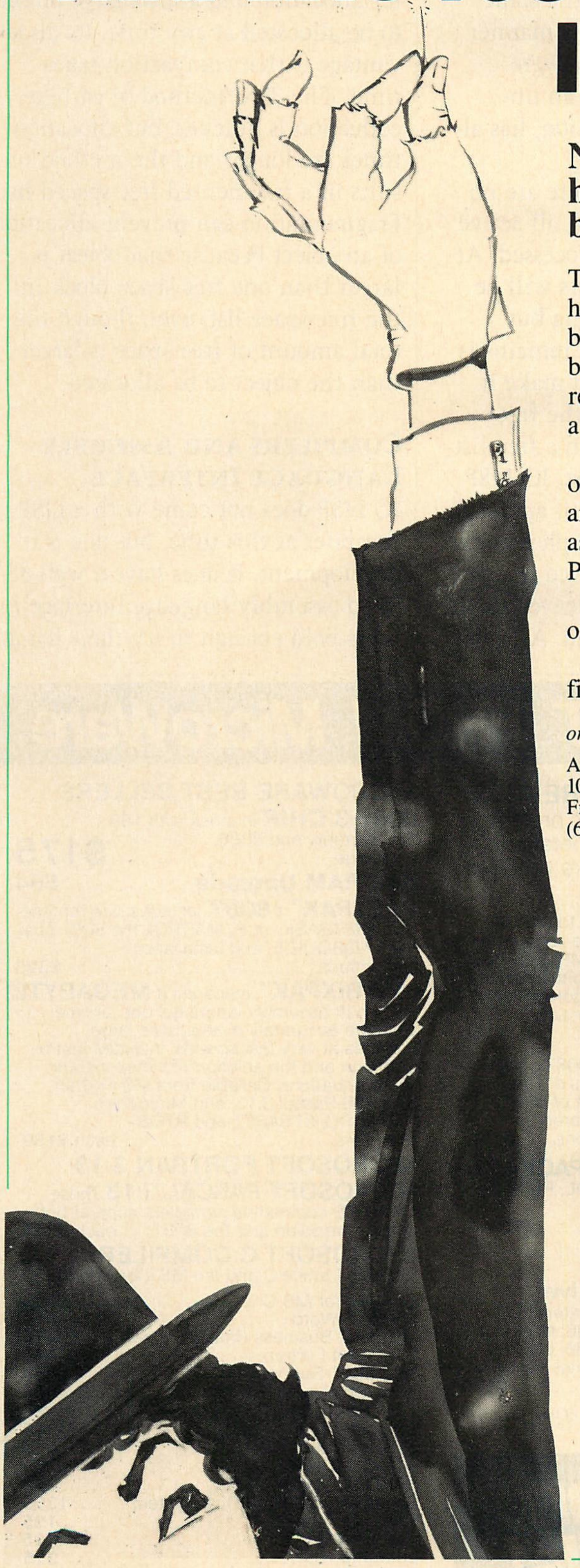
The muLISP error-handling, or lack thereof, is reminiscent of earlier LISP implementations. There are few errors that can occur, but most of these are fatal. Those from which the system can recover cause it to start again at the top level; erroneous computation is thus lost. For example, the result of evaluating a function with no definition is the list being evaluated. No error occurs in this case in muLISP, whereas both IQ LISP and TLC LISP would recognize this as an error condition. Another example is that muLISP does not care if the number of defined parameters matches the number of actual parameters. If there are too many defined parameters, the extra ones will be bound to NIL. If there are too few defined parameters, those left undefined will be ignored. With muLISP, if the program runs, it will run very quickly. If it doesn't run, good luck trying to debug it.

muLISP has BREAK and TRACE support but no CATCH or THROW functions. The development package consists of the muStar editor, which can also load and store definitions. All in all, there is limited support.

TLC LISP provides functionality equivalent to IQ LISP's except that it

IBM PC™ OWNERS:

HARD DISK STORAGE WITH BACKUP THAT LETS YOU FIND A NEEDLE IN A HAY STACK.



Now, in a single package, hard disk storage and file-oriented backup: Alloy's PC-STOR.™

The remarkable new PC-STOR combines the latest Winchester hard disk technology with the industry's finest file-oriented tape backup. With this system you can locate, retrieve and re-position backup data quickly, precisely and easily. The PC-STOR makes recovery from awkward floppy disk or streaming backup systems a memory of the past for users of IBM or IBM-PC compatibles.

The new Alloy Winchester hard disk comes with 35 megabytes of formatted storage, is 4 times faster than the IBM-XT system, and is especially designed for ease of use. It can be purchased as the PC-DISC™, a hard disk system without backup, or as a PC-STOR, a hard disk with file-oriented tape backup.

If you already have a hard disk system, you should add file-oriented tape backup: Alloy's PC-BACKUP™.

For the ultimate in hard disk and hard disk backup systems, find out more about Alloy's advanced product line, today.

For more information, please contact your local computer store, or call or write:

Alloy Computer Products
100 Pennsylvania Avenue
Framingham, Mass. 01701
(617) 875-6100

In Europe, *contact:*
Alloy Computer Products (Europe) Ltd.
Cirencester, Gloucestershire, England.
Tel: 0285-68709 Tlx:43340

IBM PC is a registered trademark of IBM Corp. PC-STOR, PC-BACKUP and PC-DISC are trademarks of Alloy Computer Products.

ALLOY
Computer Products

CIRCLE NO. 226 ON READER SERVICE CARD

has no built-in interactive error facility like IQ LISP's dual-window development support. TLC LISP does support CATCH and THROW along with good TRACE and BREAK packages. As in IQ LISP, these packages are written in LISP and can be enhanced by the user. TLC LISP also gives the programmer access to the evaluation stack, and errors can be handled by program or user.

GARBAGE COLLECTORS

There are quite a number of different techniques that can be used to recover free space; the three systems described here use the same mark-sweep algorithm. IQ LISP and muLISP add a compaction phase after the garbage has been collected.

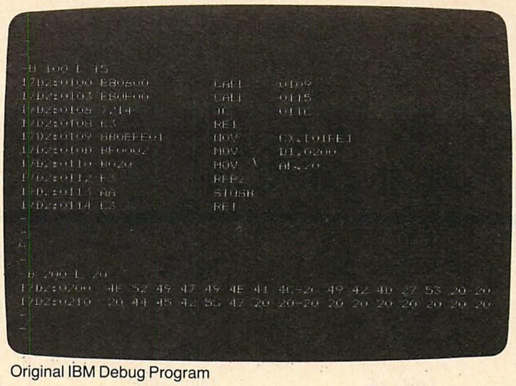
In the mark-sweep algorithm, all objects are first marked as garbage. The root, usually the active-object list, is used as the reference to start the process. The reference points to an object that is currently marked as garbage. The marker is changed since the root points only to active objects that are not to be thrown away. The object is then checked for any references, and any that are marked as garbage are processed in the same way. Objects found in this manner are processed only if they were marked as garbage, since an unmarked object, by definition, has already been processed.

Eventually, when there are no more references to follow, all active objects will have been processed. At this time, all active objects will be unmarked, leaving nothing but marked garbage. All that remains is to collect the garbage and make it free space. In TLC LISP, the free space is linked together in a free list and allocated from the list. IQ LISP and muLISP use a different approach, performing a compaction phase next. All active objects are moved to one end of the space, which leaves all the free space at the other end. All references are adjusted accordingly for proper operation. Allocation is now simply a matter of taking a chunk out of the single free-space section. Compaction has the advantage of faster allocation and allows large objects to be allocated at any time. Its disadvantage is that compaction takes time. The TLC method of garbage collection is quicker, but allocation times are longer and the method results in a fragmented free-spaced list. Fragmentation can prevent allocation of an object because that object is larger than one free-space block in the free-space list, even though the total amount of free space is larger than the object to be allocated.

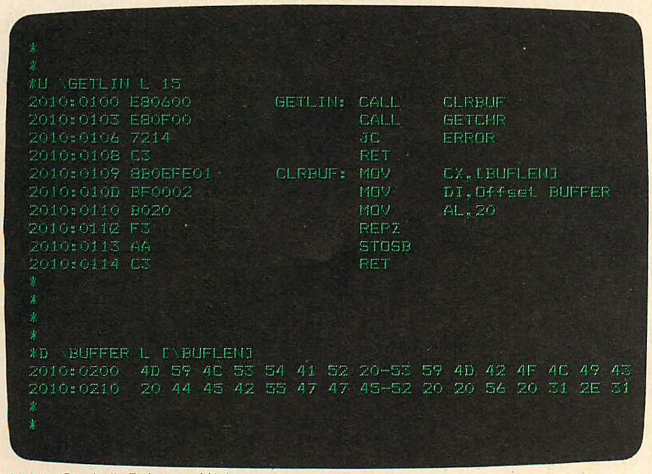
COMPILERS AND ASSEMBLY LANGUAGE INTERFACE

IQ LISP does not come with a LISP compiler at this time, but one is in development. It does have a well-defined assembly language interface for users bold enough to try their hand

The difference



Original IBM Debug Program



Mylstar Symbolic Debugger V1.1

is Mylstar's Symbolic Debugging Program*

The plain and simple difference is that Mylstar's Symbolic Debugging Program speaks to your IBM PC in a language you both can understand, plain and simple.

Employing the same command structure, it allows you to use symbol names, mathematical expressions, batch files, on-line help, multi-command macros and other time-saving entries.

It's the enhancement to the *IBM Debug Program* you've been looking for—because it fills in the gaps—shortening the frustrating debugging process by as much as 50%—leaving you more time to do the work you need to do and the work you want to do, plain and simple.

Mylstar's Symbolic Debugging Program has been programmer-tested for over a year at Mylstar Electronics, Inc., (formerly D. Gottlieb & Co.), designers of the video arcade game, Q*BERT™.

TO ORDER...
Call (312) 562-7400 or mail coupon today.

**Designed for IBM PC-DOS 1.1 with 128K RAM minimum*

CIRCLE NO. 124 ON READER SERVICE CARD

MYLSTAR ELECTRONICS INC.

165 West Lake Street
Northlake, Illinois 60164

A Columbia Pictures Industries Company

Mylstar Electronics, Inc.,
165 W. Lake St., Northlake, IL 60164

Please send me Mylstar's Symbolic Debugging Program for use with the IBM PC computer. Enclosed is $125, plain and simple.

☐ Check ☐ Money Order 113

NAME

FIRM

ADDRESS

CITY STATE ZIP

Illinois residents add 7% sales tax
Allow 2-4 weeks for delivery

Table 2:
Comparison of IQ LISP, muLISP, and TLC LISP

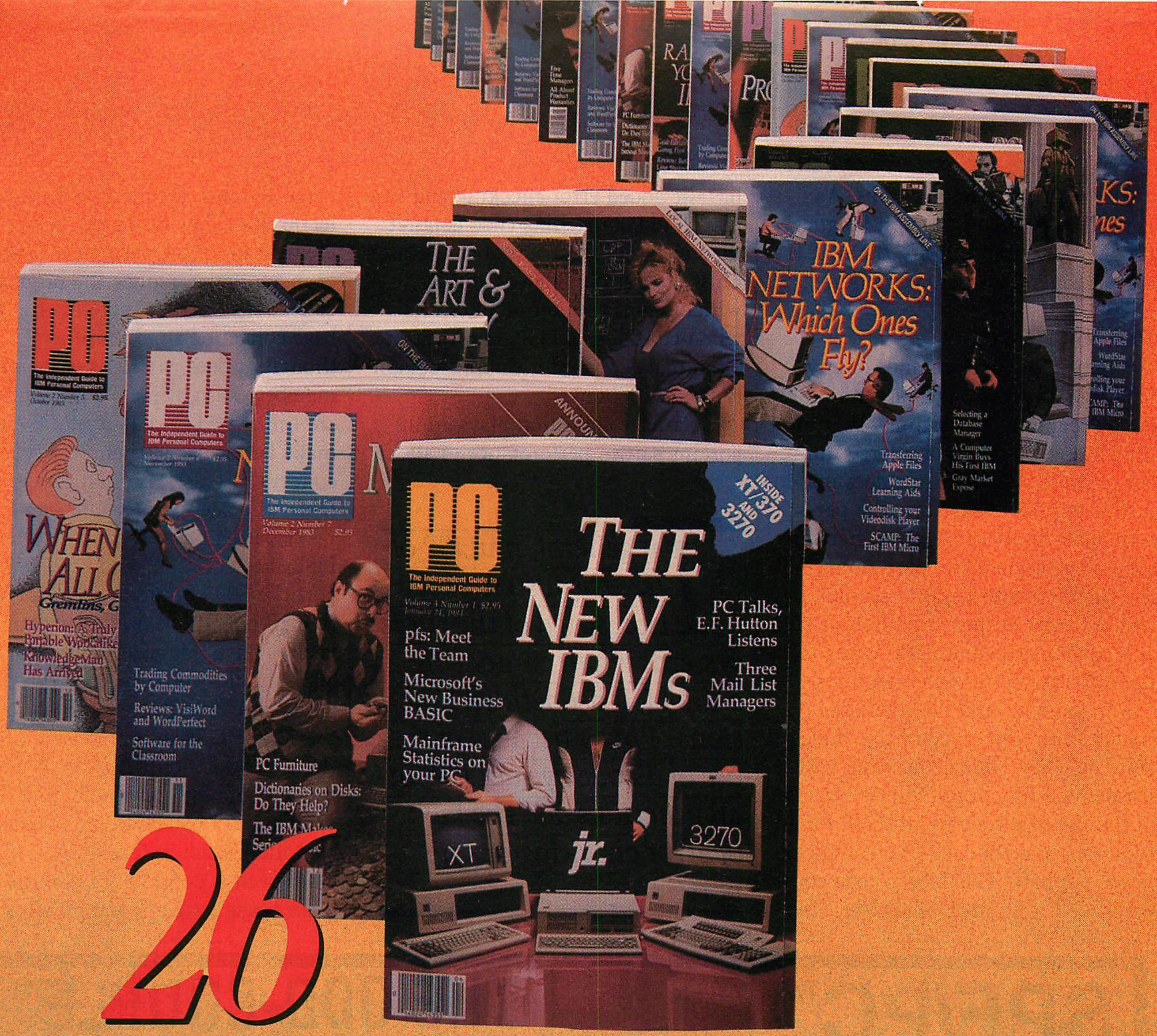
	muLISP-82	IQ LISP	TLC LISP
Price	$250	$175	not available
Version	3.13	1.4.5	1.26
Similar to	LISP 1.5	MACLISP	MACLISP
Functional Forms			
LAMBDA	yes	yes	yes
FLAMBDA	yes	yes	yes
MACRO	no	yes	yes
Classes	no	no	yes
Error-Handling			
General	poor	very good	good
BREAK	no	yes	yes
CATCH/THROW	no	yes	yes
Debugger	no	screen	yes
Argument check	no	no	yes
Editor	screen	line	screen
Functions Provided			
Library	good	excellent	excellent
Documentation	poor	good	good
Sequential I/O	poor	good	very good
Random I/O	no	no	good
Console Support			
Function keys	poor	very good	good
Windows	no	scrollable	non-scrollable
Graphics	no	line/dot	turtle
Memory Limits			
List Space	256K bytes	1 Mbyte	256K bytes
Data Space	none	1 Mbyte	1 Mbyte
Total Space	256K bytes	1 Mbyte	1 Mbyte
Objects			
16-bit integer	no	yes	yes
32-bit integer	no	yes	yes
Big integer	yes	yes	no
Floating point	no	4 or 8 byte	4 byte
Strings	32K bytes	32K bytes	32K bytes
Vectors	no	array	yes
Arrays	no	yes	no
Machine Interface			
Machine code	no	yes	in development
D-code	yes	no	no
P-code	no	no	yes
OS Functions	no	yes	yes
8087 support	no	yes	yes
Operating System			
PC-DOS V2.0	yes	yes	no
CP/M-86	yes	no	yes

at it. The documentation describes the interface in detail, and a complete example is included. Assembly language routines can greatly increase the speed of execution and can be freely mixed with normal LISP definitions. It is possible for a user to write his own LISP compiler, but it would be quite a job.

muLISP uses a built-in compiler for all LISP function definitions. It generates d-code (distilled code), which is similar to the list syntax but is more compact and runs faster. The original LISP list format can be recreated using the built-in decompiler. Both operations are transparent to the user, since the system performs them automatically. In fact, there is no way to directly examine or modify the d-code. There is also no description of the internal format of the d-code in the documentation. No method is provided for adding assembly language routines to the system.

TLC LISP is again somewhere between IQ LISP and muLISP. Typical LISP definitions use the normal LISP list format. There is also a LISP p-code interpreter, assembler and compiler. The decision to use p-code must be made by the user, and functions must be processed using the compiler. The documentation describes the p-code and interpreter operation, so it is also possible to use the assembler. The examples given are sufficient for anyone familiar with LISP and assembler. P-code runs two to three times faster than list-format definitions, but not as fast as machine language. A compiler to change LISP to machine language is in the works but is not currently available. Also, the documentation does not currently give any description of the machine-language interface.

LISP compilers are the next big hurdle on the IBM PC. If mainframe results are any indication, we can look forward to some very good native code (machine language) compilers. Good LISP compilers are known to generate code that rivals that of optimized FORTRAN and PL/I compil-



26 REASONS TO SUBSCRIBE TO PC

Your subscription will bring you 26 big issues every year of the most up-to-date facts and technological breakthroughs that affect you and your IBM or compatible personal computer. 26 big issues of *PC* delivered to your door AT UNBELIEVABLE SAVINGS!

Every other week you'll get a library of professional insights including latest coverage of newly-released hardware and peripherals. You'll find timely information on innovations in programming, applications and networking. You'll communicate with professionals and users like yourself and get reviews of new software packages, some before they even hit the shelves.

You can subscribe to *PC*, The Independent Guide To IBM Personal Computers for as little as $1.02 an issue! Use the attached card or the coupon at right TO SUBSCRIBE TODAY!

GET THE MOST UP-TO-DATE INFORMATION FOR YOU AND YOUR PC WITH *PC*, 26 TIMES A YEAR!

26 **SUBSCRIBE TO PC** 26

P.O. BOX 2443 • BOULDER, CO 80322

YES, Please enter my subscription to *PC* for:

☐ 26 issues (one year) for only $29.97...LESS THAN $1.16 A COPY!
☐ 52 issues for only $52.97...LESS THAN $1.02 A COPY!

Full one-year subscription price $34.97. Single copy price $2.50.

Mr./Mrs./Ms.____________________
please print name in full

8H625

Company____________________

Address____________________

City__________ State__________ Zip__________

Check one: ☐ Payment enclosed ☐ Bill me later

Charge my: ☐ American Express ☐ Visa ☐ MasterCard

Card No.__________ Exp. Date______

Add $52 a year for postage to Canada and all other foreign countries. Please allow 30 to 60 days for delivery of first issue.

ers for numerical applications. As LISP itself is one of the best languages for symbolic applications, the mix is a good general-purpose compiler.

PERFORMANCE

Evaluating the performance of any language is difficult at best, and LISP is no exception. Unfortunately, running a few DO loops or a large number of multiplications just does not do justice to LISP. Therefore, the performance observations given here are somewhat vague. The performance of the three systems is fairly similar, and one may outperform another on any particular benchmark owing to the different system architectures.

In general, muLISP offers the fastest execution speed of the three systems when comparing basic LISP list defined functions. This is because of the use of d-code by muLISP for all functions and the lack of error checking or support. TLC LISP p-code compares well with muLISP d-code, but has the added error-checking and support. IQ LISP machine code is faster than either of the above, but the user must program in 8086 assembler, which seems counter-productive when considering LISP. However, it often makes sense for small, frequently used functions.

Garbage collection plays an important part in performance of all three systems. Frequent garbage collection can slow down the fastest implementation. Increasing the amount of memory can reduce the number of collections but also increases the garbage collection time. In general, though, it does pay to have the extra memory. Also, if memory resources are limited TLC LISP or muLISP will make better use of them. Garbage collection is also faster on these systems, as object pointers are smaller.

In general, muLISP offers the fastest execution speed of the three systems when comparing basic LISP list defined functions. This is because of the use of d-code by muLISP for all functions and the lack of error checking or support.

Floating point is supported by IQ LISP and TLC LISP, and both can support the 8087 numeric processor. Although operations on floating-point vectors have not been optimized, the 8087 should work especially well when accessing this type of object.

These LISP interpreters run faster than most of the interpreter BASIC implementations on the PC and slower than most conventional language compilers, such as C and PASCAL. The speed is dependent upon the program being tested; it is difficult to

Ferocious FORTRAN.

Microsoft® FORTRAN crunches numbers with a vengeance!

It combines fast and efficient native code compilation with built-in 8087 coprocessor support. The result? Mini and mainframe performance from your MS™-DOS micro.

Based on the '77 standard, Microsoft FORTRAN supports extensive statements and data types—including complex numbers and IEEE single and double-precision floating point accuracy.

Support for large arrays (greater than 64K bytes), separate module compilation, and overlays, allow you to create very large programs—up to one megabyte, with access to more than 65 thousand records in a file as large as four gigabytes.

MICROSOFT.
The High Performance Software

How do programmers feel about Microsoft FORTRAN?

"The first FORTRAN compiler that takes advantage of the full addressing capability of the 8088 and the power of the 8087."
—Jack Wilschke, *Softalk*

"We decided to use the Microsoft FORTRAN Compiler for its INTEGER 4 capability and the flexibility of its 8087 implementation."
—Charlie Huizena & Chip Barnaky, *PC World*

Call 800-426-9400 to order the ferocious FORTRAN. $350.*

In Washington State, call 206-828-8088. Ask for operator C4, who will rush you your order, send you more information, or give you the name of your nearest dealer to see Microsoft FORTRAN in action.

*Price exclusive of handling and Washington State sales tax.

compare speeds when AI-type problems are concerned, primarily because of the difficulty in programming the problem in more conventional languages like C and PASCAL.

SUMMARY

For a capsulized comparison of the three systems, see table 2. TLC LISP is the best LISP on the IBM PC under CP/M-86 or one of its cousins; IQ LISP is the best under PC-DOS. Unfortunately, neither TLC LISP nor IQ LISP runs under both operating systems, so there is no real competition between the two. muLISP provides a respectable LISP system that runs under both PC-DOS and CP/M-86.

TLC LISP and IQ LISP provide good compatibility with many large mainframe LISP systems for people considering possible upgrade paths. TLC LISP and muLISP are available on 8-bit CP/M systems for anyone considering movement between 8- and 16-bit systems.

LISP is quickly moving into the marketplace. The popularity of the IBM PC; the availability of three good LISP systems; and the PC's large address space, which allows development of significant applications using LISP, should help to extend the popularity of the language; LISP is no longer restricted to the large mainframes or to the computer scientist.

PRICES AND AVAILABILITY

IQ LISP V1.4.5
Integral Quality
P.O. Box 31970
Seattle, WA 98103-0070
(206) 527-2918
Price: $175
CIRCLE 499 ON READER SERVICE CARD

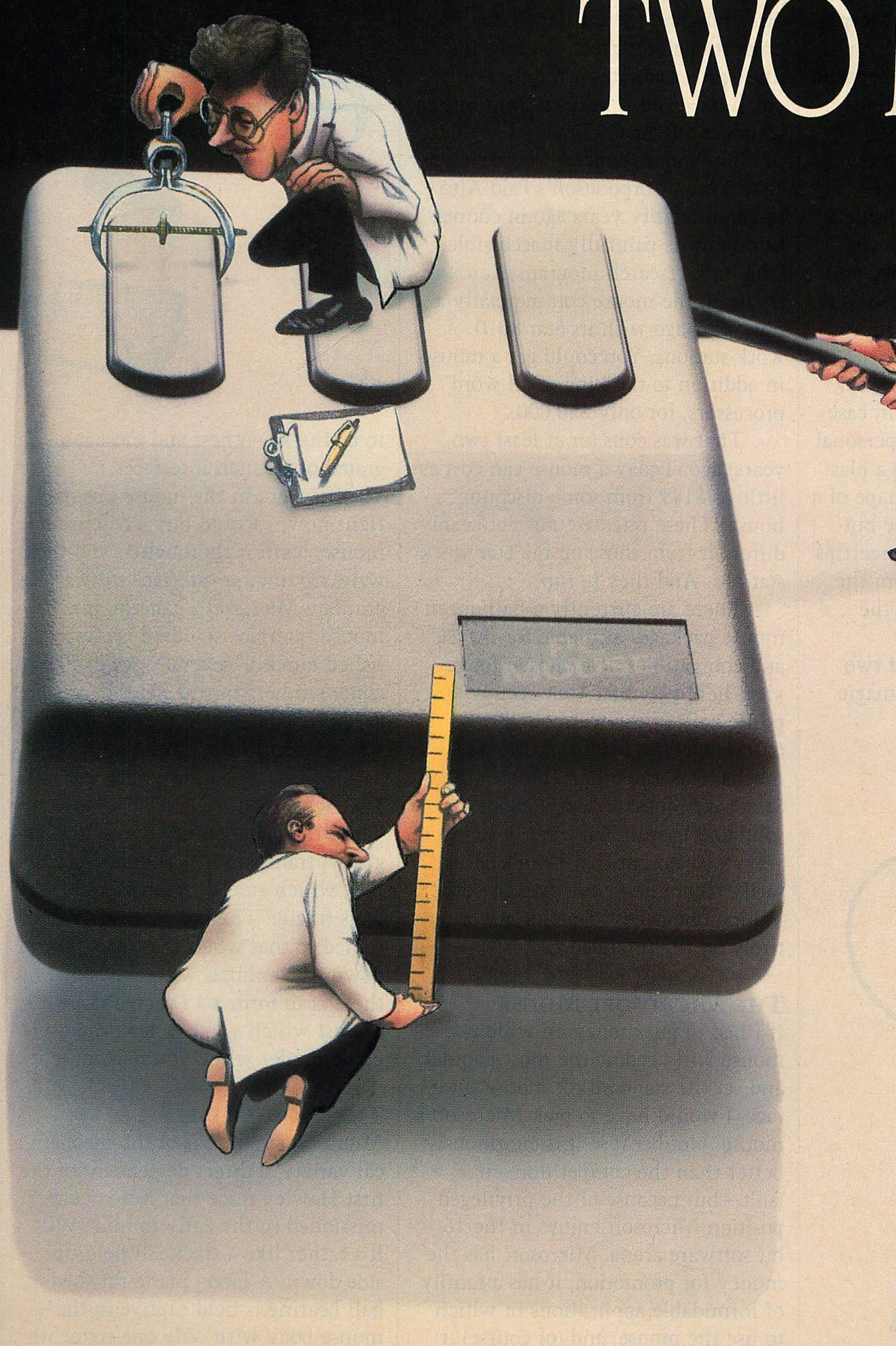
muLISP/muStar-82
The Soft Warehouse
P.O. Box 1328
Honolulu, Hawaii 96828
808-734-5801
Price: $250
From: Microsoft Corp.
101700 Northup Way
Bellevue, WA 98004
206-828-8080
CIRCLE 497 ON READER SERVICE CARD

TLC LISP
The LISP Company
330 C Village Lane
Los Gatos, CA 95030
408-354-3668
Price: Not available
CIRCLE 496 ON READER SERVICE CARD


MICROSOFT

A TALE OF TWO MICE

JEFF DUNTEMANN



Microsoft's mechanical mouse and Mouse Systems' optical mouse: how do they fare in the electronic maze?

I am obsessed with a desire to point to things. I don't want to *tell* you where it is. I don't want to give you an address or an *X, Y*. I don't want to batter my poor cursor keys until they crack off and fly into my lap. I want to move my hand and zero a cursor onto a target like *that*. There are gadgets that point. I've tried them all. I dislike light pens; light pens dislike my long-persistence phosphor monitor. Joysticks are weird and

nonlinear and just too fast; one muscular tick on the stick and your cursor is ten miles west of Poughkeepsie and still traveling. Bit pads are, well, bit pads. If I have to hold the Platypus Pad and press its buttons with one hand and scribble with a popsicle stick in my other hand, I might as well just use the keyboard.

Hewlett-Packard's slick touch screen is certainly the direct approach, but my fingers are considerably more than one pixel wide, and the combination of wood smoke and peanut butter that usually prevails at my house would make for opaque screens in no time.

None of these approaches works for me: what I like are mice. In case you only recently joined the personal computing family, a mouse is a plastic brick about the size and shape of a bar of soap. It has two or three buttons on the top where your fingertips rest and a cable extending from the back to connect the device to the back of the computer.

Inside the mouse is one of two different flavors of high-tech magic

that translate motion on a tabletop to motion on a screen. Slide the mouse around on the tabletop; up on the screen the cursor moves in the same direction, and for a distance that *feels* proportionally correct. It's fast. It's linear. It's accurate. It requires only one hand. And it keeps peanut butter smudges off the CRT.

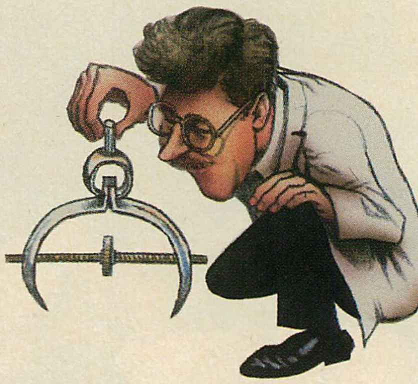
The mouse concept was developed at Xerox Corporation's Palo Alto research facility years ago in connection with its painfully inaccessible Smalltalk research program. Xerox introduced the mouse commercially a few years ago with its Star 8010 work-stations. You could get a mouse, in addition to the king of all word processors, for only $50,000.

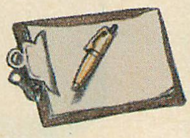
That was eons (or at least two years) ago. Today a mouse can cost as little as $149 from some discount houses.These mice are not noticeably different from those on big Star workstations. And they're *fun*.

There are currently a half-dozen mouse packages available for the PC and compatibles. I've chosen for review here two mice that occupy opposite ends of the technology spectrum. One, the Microsoft Mouse, is mechanical; the other, the Mouse Systems Mouse, is optical. Both are solid and useful. They are in many ways so different that I can't necessarily recommend one over the other. Which one is best depends almost entirely on the intended use.

THE MICROSOFT MOUSE

If I had to put money on which mouse will become the most popular and set the standard for mouse interface, I would have to pick Microsoft's mouse. Not because it is enormously better than the competition—it isn't—but because of the privileged position Microsoft enjoys in the 16-bit software arena. Microsoft has the money for promotion, it has a family of formidable applications in which to use the mouse, and (of course) it owns the most important 16-bit operating system. Microsoft's mouse interface to MS DOS and PC DOS is likely





to be the smoothest and most integrated of all such interfaces.

If I were in the mouse business right now, I would buy a Microsoft mouse, learn it thoroughly, and then write my mouse interface software to emulate Microsoft's *exactly*. If a mouse interface standard is established quickly, several mice can become winners in this field. If no standard is established, only Microsoft will remain in the mouse business.

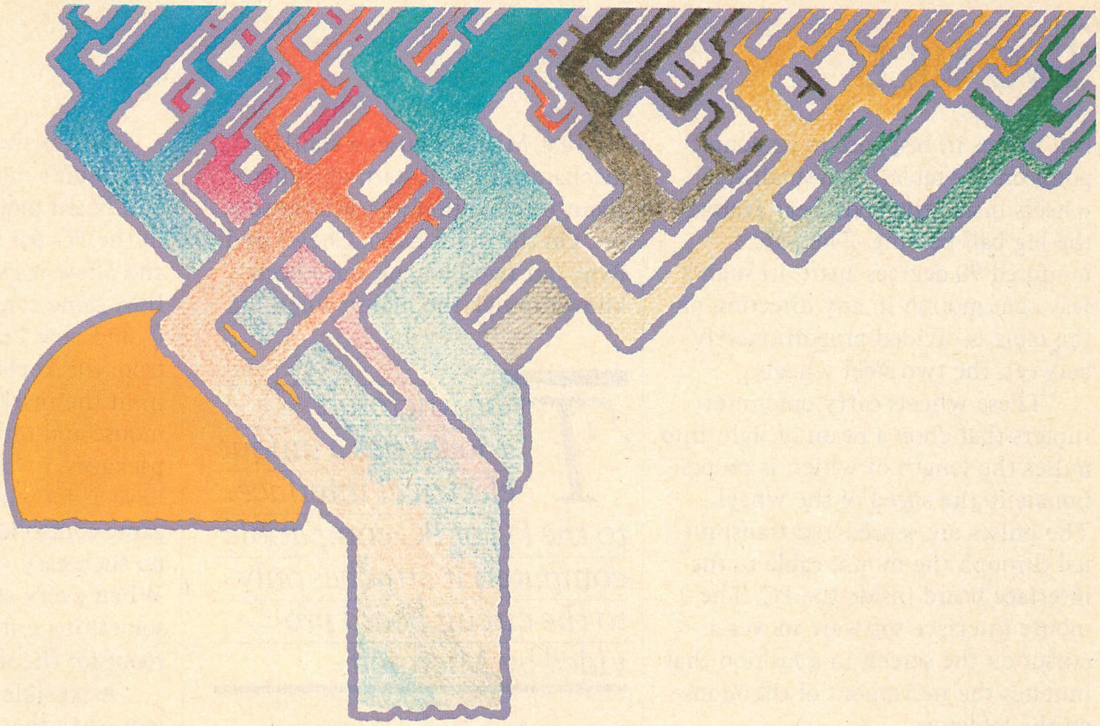
PACKAGING

The Microsoft Mouse comes in a glassy, transparent flip-top styrene box, which smells expensive and feels fragile. The box is cleverly designed so that when the top is flipped down, it combines with the body of the box to form a 45-degree slope against which you can lean the user-guide binder while you're reading it.

THE MOUSE ITSELF

Microsoft's mouse is of the mechanical variety, a direct descendant of the first Hawley mice that Xerox commissioned in the early to mid 1970s. It's rather like a trackball held upside-down. A large, matte-finished ball bearing is held captive in the mouse body with only one-sixteenth inch of ball protruding through a circular hole at the body base. As the mouse slides around on the table the

ILLUSTRATION • MIKE OKAMOTO



ball moves in response to its contact point on the table. Two small steel wheels inside the body bear against the big ball bearing. They are mounted 90 degrees apart, in such a way that motion in any direction on the table is divided proportionately between the two steel wheels.

These wheels carry opto-interrupters that chop a beam of light into pulses the length of which is proportionate to the speed of the wheel. The pulses are sensed and transmitted through the mouse cable to the interface board inside the PC. The mouse interface software moves a cursor on the screen in a fashion that imitates the movement of the mouse on the tabletop.

The mouse body is constructed of a beige plastic that looks as if it could handle a fair amount of abuse. Although I have not yet removed the bottom plate to examine the mechanism, I have seen a Xerox mouse stripped naked, and its works looked complicated and fragile. I am assuming that Microsoft uses a similar mechanism; I will start worrying about its condition in a year or so.

On the other hand, Chewy and Byte, our two Bichons, have been known to lick the mouse when I'm not using it, and so far the puppy-slurp has not had any adverse affects. The mouse *does* pick up fuzz and doghair from the environment, especially once the ball gets tacky from sliding across the multitude of Mountain Dew rings on the desktop. It is possible, by removing one phillips-head screw, to drop the ball out of the mouse body and wash it. Possible, and a good idea, probably on a monthly basis. Dry the ball with care—I suspect rust would make it seize up inside the mouse body. And check inside the ballrace for other miscellaneous accumulations.

On the mouse's back are two green bar switches that click satisfyingly when pressed. The mouse as a whole is sculpted to fit the hand reasonably well. I am generally pleased with the physical design.

The Microsoft Mouse has a unique electrical interface to the PC or PC-compatible computer; it attaches only to the circuit board provided by Microsoft.

TABLE 1: Microsoft Mouse Driver Function Calls

Function Call #	Description
0	Return Installed Flag; reinvoke default values
1	Display cursor
2	Hide cursor
3	Poll mouse position and button status
4	Set mouse cursor to absolute *X/Y* position
5	Poll button-press information
6	Poll button-release information
7	Set minimum/maximum horizontal cursor travel
8	Set minimum/maximum vertical cursor travel
9	Load new graphics cursor pattern & "hot spot"
10	Set characteristics of text cursor
11	Read mouse motion counters
12	Set user subroutine mask and address information
13	Turn light pen emulation feature on
14	Turn light pen emulation feature off
15	Set mickey/pixel ratio

The Interface Board

The Microsoft Mouse has a unique electrical interface to the PC or PC-compatible computer. In other words, it does *not* attach to a serial or parallel port; it attaches only to the circuit board provided by Microsoft. This circuit board is tiny, and yet it contains all of seven integrated circuits and a few other components. As far as I can determine (Microsoft does not supply the schematic), it is nothing more than a pair of parallel ports.

This is the most serious defect in Microsoft's mouse package. The circuit board monopolizes an entire slot in the PC, for the sake of seven IC's and a few resistors. I had to yank my IBM game controller to find room for it, and now I'm forced to play J-Bird from the keyboard. This is going to limit the market for Microsoft's mouse and ultimately its application packages. I was fortunate enough to have a slot with nothing more than a game controller in it, but others have no such easy yank in their machines. When every slot is already doing something critical, how can you find room for the mouse?

A possible solution to this problem—one that worked beautifully for IBM—is for Microsoft to publish schematics and full interface specs for its interface card, allowing multifunction board vendors to incorporate the relatively simple circuitry on the latest generation of 23-function PC add-on boards. After all, Microsoft is out to sell mice, not circuit boards. Once lots of people have multifunction boards with mouse ports, Microsoft can drop the circuit board and sell the mouse alone at a lower price that would stimulate greater sales.

The mouse cable plugs in to the circuit board through a DB-15 connector. Unlike most serial port cables, the mouse cable connector lacks the little captive side screws that allow a firm connection to be made between cable connector and circuit board. Thus the mouse cable pulls out of the circuit board very easily. This is a bad idea electrically and could conceivably damage either the mouse, the board, or both. (By the way, if the cable does pull out, turn the PC off before plugging the mouse back in again.) The omission of these screws is unfortunate.

Performance

Those considerations aside, the Microsoft Mouse is one fine piece of gear. In addition to heavy testing on the IBM PC, I installed the mouse for an evening in a friend's Compaq and

found no differences in operation. The interface software is a single program set up as an "exit & remain resident" module that loads at the bottom of the TPA and resets DOS's TPA pointers so that the next program to load does so *above* it. The program is run once at boot time (usually as part of an AUTOEXEC.BAT batch file), and it loads and remains usable in memory until the next time the machine is booted. It is about 2500 bytes long, not at all large for what it accomplishes.

The software monitors the mouse for motion and button activity. It maintains an *X/Y* position for the mouse on a virtual 640X200 grid, plus information on whether the two buttons are up or down. It also tallies the number of times the mouse buttons have been pressed between calls to the driver software. Additionally, the driver software maintains on the screen a cursor that moves in response to the mouse, regardless of what other process is running on the machine. It is interrupt-driven and efficient enough not to slow down the "foreground" process noticeably.

There are 14 function calls to the driver software (see table 1). These function calls may be made easily

Despite some problems, the Microsoft Mouse is one fine piece of gear. In addition to heavy testing on the IBM PC, I installed the mouse for an evening in a friend's Compaq and found no differences in operation.

from BASICA or Microsoft's compiled languages. I have yet to think of any important function not included, though I might alter how one or two of them work.

Calling the functions from BASICA is easy enough. Once the entry point to the mouse driver is located (which is as easy as finding the interrupt vector for software interrupt 51) and stored in a variable (for example, MOUSE), a mouse driver call is made like this:

```
CALL MOUSE (M1%,
  M2%,M3%,M4%)
```

The first parameter is the function call number. The remaining parameters have various functions (or no function at all) depending on the function call number.

Calling the mouse driver from Microsoft's compiled languages (COBOL, FORTRAN, Pascal, and the BASIC Compiler) is done by linking in routines from an included library called MOUSE.LIB. Assembly code can call the mouse driver by pushing parameters on the stack and executing software interrupt 51.

Most of the function calls are self-explanatory, but several could use some clarification. Calls 5 and 6 allow a programmer detailed information on button press and release, including the number of presses and releases since the last call to the function, and the *X/Y* position of the mouse at the time the last press or release was noted.

Calls 7 and 8 confine the mouse cursor to a given range of screen coordinates. Used together, they allow a programmer to keep the mouse cursor within a rectangular window of any desired size for selection of items in that window. Call 9 allows the programmer to change the shape of the graphics cursor by loading a 32-byte pattern to become the new cursor shape. The manual includes a compendium of cursor shapes, including a pointing hand, various arrows, and an hourglass. The "hot spot" is the cursor pixel from which the mouse driver derives the mouse's *X/Y* position. It can be any pixel within the 16×16 grid.

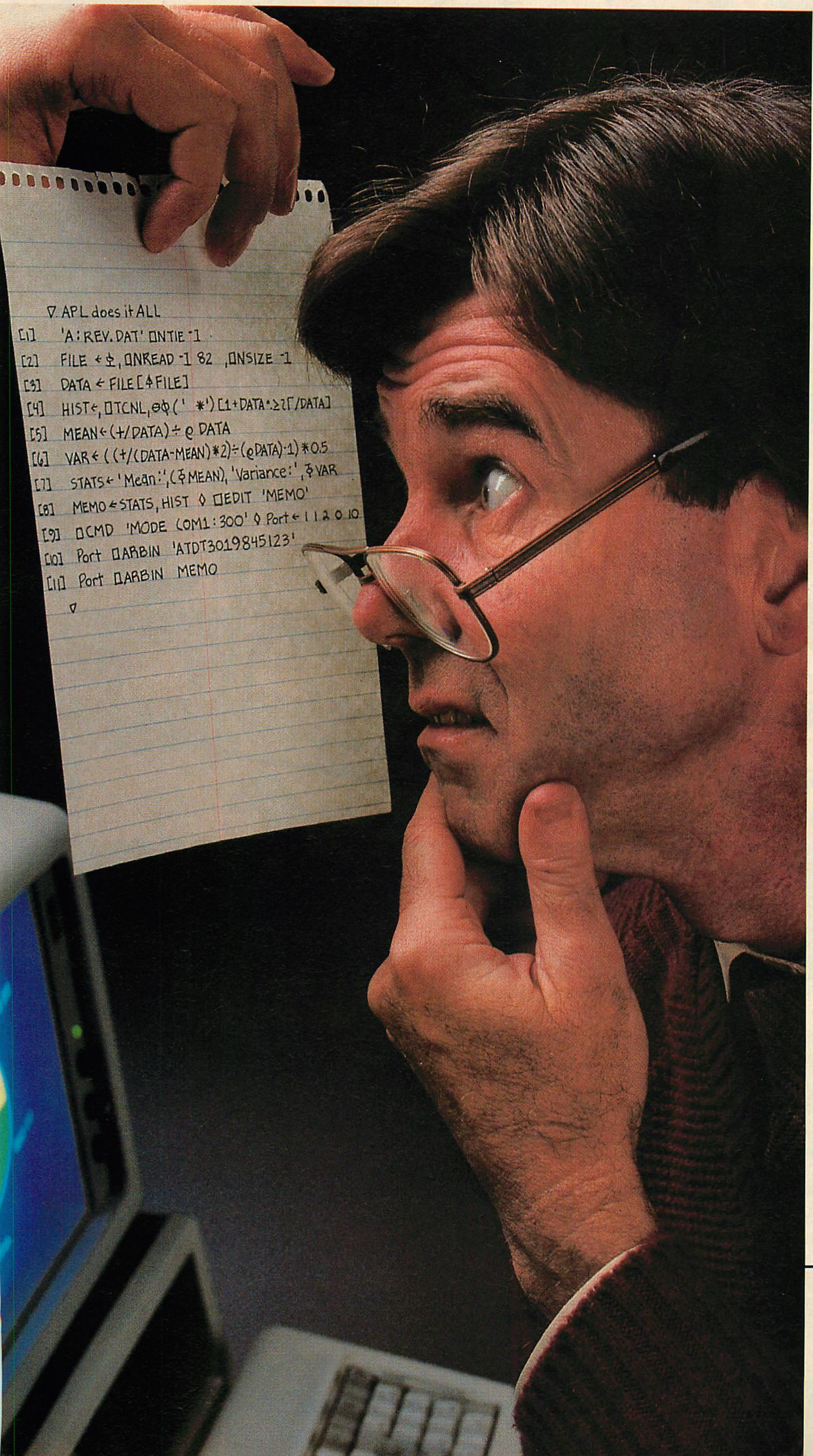
Call 10 returns the number of "mickeys" (no, I'm not kidding) through which the ball bearing has rolled since the last call to this function. A mickey is one "tick" on the optical interrupters in the mouse's mechanism; it corresponds to 1/100" movement of the ball. The driver keeps track of mickeys rolled independently in both the horizontal and the vertical direction.

Function 12 is perhaps the cleverest of all: it allows a programmer to hand the mouse driver an address of a machine-language subroutine, along with a bit-map mask that controls the circumstances under which that subroutine is called.

Function 12 is perhaps the cleverest of all: it allows a programmer to hand the mouse driver an address of a machine-language subroutine, along with a bit-map mask that controls the circumstances under which that subroutine is called. One bit on the mask corresponds to each of the two buttons; another corresponds to any change in cursor position. If a bit in the mask is high, the machine-language subroutine will be called whenever that bit's condition is met. For example, if the bit in the mask for cursor-position change is raised, the subroutine will be called each time the cursor position changes. The obvious application for this function is a fast pixel-draw routine.

Call 15 allows programmer control of mouse cursor speed relative to actual mouse speed over the table.

Tell this to your PC and see what happens.



No, it won't roll over and play dead. While the funny-looking symbols might not make immediate sense to you, to a PC equipped with STSC's APL★PLUS®/PC System they tell an amazing story. In just 11 lines this program describes a sequence of events that can't be accomplished by any other single software package.

The program is written in an application development language called the APL★PLUS/PC System. Briefly, here's what it does. First, it sorts a DOS file containing revenue data and plots the revenues as a histogram. It calculates mean and variance revenues. Then with the help of a full-screen editor, it creates a memo combining the histogram, statistics and descriptive text. Finally the program issues a DOS command to the PC, dials a host computer, and electronically mails the memo. All in just 11 lines. No wonder a *PC Magazine* reviewer reacted to our APL★PLUS/PC System with "awe and delight" (March 1983).

If you want to integrate existing software or create custom solutions to problems that can't be handled by the software you're currently using, you need the APL★PLUS/PC System. It costs $595.00, and runs on the IBM PC with 192 KB of RAM as well as on a number of compatible machines.

To order the APL★PLUS/PC System or to receive your free convincer kit, write or call STSC, Inc., 2115 East Jefferson Street, Rockville, Maryland 20852. (301) 984-5123.

CIRCLE NO. 208 ON READER SERVICE CARD

STSC
A Contel Company

This speed can be set independently for horizontal and vertical travel, which is a convenient way to compensate for the higher-than-wide pixels on the PC: default ratios are 1 mickey per pixel horizontally and 2 mickeys per pixel vertically. The actual amount of desk space required to move the cursor around the entire screen, using these default values, is 6.4 inches by 4 inches.

The mouse driver is fast. There is no apparent lag between the motion of the mouse and motion of the cursor. Individual function calls are executed extremely rapidly, enabling me to write a very satisfactory mouse-based graphics-generator program in non-optimized, heavily commented interpreted BASICA. As the BASIC compiler does not yet support many graphics features from BASICA 2.0, I will have to be content with the interpreter. So far performance has been more than adequate.

DOCUMENTATION

The user guide is nicely typeset and of a professional quality. Instructions for installing the mouse interface board and mouse software are painstakingly complete and crisply illustrated in the fashion of IBM's manuals. Rank beginners might have trouble understanding the interface specifications, but then again, they are unlikely to need to understand them. The portions of the manual a beginner would need are slanted toward beginners. I suspect a newcomer to computing will have no trouble here if he reads the text.

The technical information on mouse interface is complete up to and including interface to Microsoft's high-level languages. I wanted a little more detail, since I intend to call the mouse from Pascal/MT+ and various Modula 2 compilers, none of which come from Microsoft. Assembly language interface is given, so it is possible to work backwards from that, but a detailed description of what the mouse driver is actually doing would have been very welcome.

PROBLEMS

I have identified one bug in the driver so far, and it is not especially serious. Functions 7 and 8 allow you to "trap" the mouse cursor within a given rectangular area of the screen, largely to support windowing. Function 0 is supposed to reset all the mouse parameters to the defaults, including mouse travel over the full 640X200 virtual screen. Unfortunately, function 0 does not "free" the mouse cursor from the limits established by calls 7 and 8. Another pair of function calls to 7 and 8 must be executed to set mouse cursor travel back to the full screen. Aside from the additional code bulk, this is not a

problem, given the speed with which the Microsoft Mouse driver executes function calls.

The only other defect is a tendency of the switches not to "catch" when pressed. Occasionally the mouse driver seems not to notice when a button is pressed. I'm not certain if this means the switches are marginally adjusted, badly made, or if some fluke in the driver software ignores the switches during some critical portion of the hardware interrupt processing cycle, thus "losing" a keypress now and then.

I spoke to people at Microsoft about these problems, but got no satisfactory explanation.

DEMO SOFTWARE

Three demo programs come with the Microsoft Mouse. One is the classic Game of Life, mouse-style. The Game of Life is a peculiar little software toy giving you a field of little bricks that may be on or off. An algorithm loosely based on bacteria culture dynamics governs whether bricks are "born" (get turned on) or "die" (get turned off). When bricks get too crowded, they die. When they become too lonely, they die. (Sounds like tropical fish.) With just the right number of chums, they just sit there and live, or "give birth" to an adjacent cell. The player's job is to clear the field, scatter an initial population of bricks across the field, then crank up the generations and see what happens. Recognizable patterns appear, flow, and change on the screen. Weird little creatures called gliders dogwalk across the screen. Weird little creatures called gliders dogwalk across the screen, and big diamond shapes fold and unfold gloriously before collapsing into small squares that flash like railway semaphores. The mouse is handy for laying down bricks in prearranged patterns, to "stack the deck" prior to letting the generations roll. Good fun for ten minutes or so.

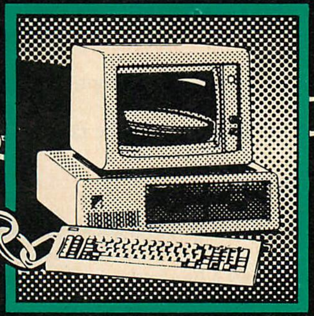
The second demo program is a piano simulator that puts piano keys on the screen. When the mouse cursor hits a key, the key plays a note. Simple and effective. Kids love it.

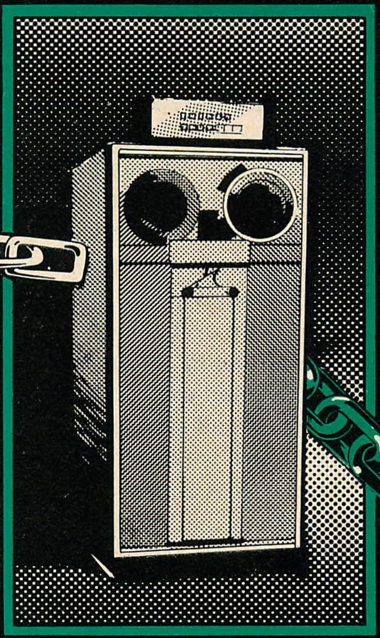
The third demo is a version of MultiTool Word called MultiTool Notepad. It is supposed to be a tool for taking notes, but the overhead involved in learning a new word processor, if you don't already know MultiTool Word, is just too much to go through simply to take notes. MultiTool Notepad works well, however, and is an excellent demonstration of the uses of the mouse for text applications rather than graphics.

TO SUM UP

The Microsoft Mouse will undoubtedly be the standard. With some luck there will be better mice sold by

Does your MICRO-TO-MAINFRAME LINK look like

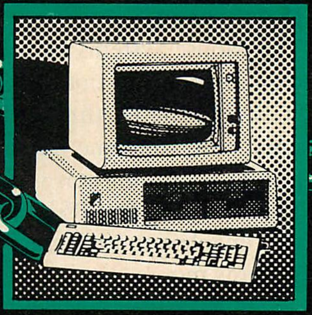




this?

hundreds of companies have installed TEMPUS-LINK because it provides:

- ☐ **simple** file transfers such as PC/DOS* Copy
- ☐ **better performance** than batch communications
- ☐ **executive friendly** for all PC/DOS applications
- ☐ **independence** from the communications devices
- ☐ **distributed processing** from PC/DOS applications
- ☐ **continuity** by adhering to PC/DOS standards



- ☐ TEMPUS-LINK micro operates under PC/DOS (or MS/DOS**) on IBM-PC* and many other compatibles.
- ☐ TEMPUS-LINK mainframe operates in CICS*, IMS*, TSO*, CMS* under DOS/VSE*, OS/VS1*, MVS*, VM*.
- ☐ TEMPUS-LINK supports major communication devices in 3270 and Asynch modes.

For information contact:

MICRO TEMPUS INC.
440 Dorchester Blvd. West, Suite 300
Montreal, Quebec H2Z 1V7

Phone: 514-397-9512
Telex: 055-61829
U.S.: (800) 228-LINK

MICRO TEMPUS INC.

*Trademarks of International Business Machines Corp.
**Trademark of Microsoft Corporation

CIRCLE NO. 272 ON READER SERVICE CARD

more responsive manufacturers adhering to that standard, but Microsoft's is, for now, satisfactory.

The Mouse Systems PC Mouse

The optical mouse is another breed entirely. Optical mice have no moving parts. They "see" where they are going with photosensors. Without the need for a rolling ball or freely turning wheels touching the ball, optical mice tend to be more reliable, especially in dirty environments. A light swabbing with a Q-tip over the optical windows is all the maintenance an optical mouse should ever need.

Perhaps the best and most popular of the optical mice is the Mouse Systems PC Mouse. It is almost completely different from the Microsoft Mouse, in both good and bad ways. Mouse Systems PC Mouse is a fine product—if, at $295, a slightly expensive one. I tested it on both the IBM PC and the Compaq, and Mouse Systems assures me it runs on the Columbia MPC as well.

Packaging

Mouse Systems packs its mice in cardboard boxes, nestled in a diecut block of foam. I prefer this to Microsoft's more expensive packaging.

The Mouse Itself

All mice are motion digitizers. They translate motion into pulses that are counted and acted upon by driver software in the host computer. Optical mice work by sliding around on a special mirrored pad, upon which has been printed a two-color grid. The vertical grid lines are blue, and horizontal grid lines are red or infrared. (Can a line be infrared? Of course! An object shows color by reflecting one color better than others. A line that reflects infrared light better than other colors is infrared.) The body of the mouse contains two sets of LED/phototransistor sender-receiver optical couplers. One uses visible light, the other infrared light.

The two beams of light strike the pad and are reflected upward. The infrared grid lines are more visible to the infrared LED, and the blue grid lines are more visible to the visible-light LED. When the mouse moves past one of the grid lines, the phototransistor that best "sees" that type of line registers a pulse with the interface electronics. Each time a pulse is received, software counters increment or decrement depending on the direction the mouse is moving. These counters are used by the interface software to position the mouse cursor on the screen and to move it in response to mouse motion on the pad.

This is how the Mouse Systems PC Mouse optical mouse works. Most other optical mice use some variation on this theme.

TABLE 2: Mouse Window Functions and Procedures

FUNCTION QUERYGRAFIX : INTEGER; Determines which graphics card is present in the PC

PROCEDURE INITGRAFIX (MODE : INTEGER); Initializes the graphics display system

PROCEDURE SETDISPLAY (PGVISIBLE, PGWRITE, PGREAD : INTEGER); Defines visible, writeable, and readable screens in multipage graphics systems

PROCEDURE VIEWPORT (LEFT, RIGHT, BOTTOM, TOP : INTEGER); Defines the active clipping viewport window

PROCEDURE SCREENPORT; Resets the active viewport to the entire screen

PROCEDURE RASTEROP (WMODE : INTEGER); Selects the raster-write mode for subsequent operations
Allows PSET, OR, XOR, NAND, and their logical inverses

PROCEDURE MOVETO (*X,Y* : INTEGER); Moves graphics cursor position to *X,Y*

PROCEDURE LINETO (*X,Y* : INTEGER); Draws a line between the cursor position and *X,Y*

PROCEDURE SCREENSIZE (VAR SCRNX,SCRNY : INTEGER); Returns the maximum valid *X* & *Y* screen values for the current graphics mode

PROCEDURE QUERYPOSN (VAR SCRNX, SCRNY : INTEGER); Returns current graphics cursor position

FUNCTION QUERYX : INTEGER; Returns current graphics cursor *X* value

FUNCTION QUERYY : INTEGER; Returns current graphics cursor *Y* value

PROCEDURE FILLSTYLE (FILLTYPE : INTEGER); Selects fill pattern for FILLPORT calls

PROCEDURE FILLPORT Fills the current viewport with fill pattern selected via FILLSTYLE

PROCEDURE WRTIMAGE (CONST BITMAP : IMAGE; WIDTHX,HEIGHTY : INTEGER); Writes array BITMAP to screen à la BASICA graphics PUT

PROCEDURE READIMAGE (VAR BITMAP : IMAGE; WIDTHX,HEIGHTY : INTEGER); Reads screen pixels into array BITMAP à la BASICA graphics GET

The Joys of Serial Interface

The Mouse Systems mouse does not require an entire slot in the PC. It "talks" to the PC through an RS232C serial port, the same way a modem or printer communicates with the PC. Theoretically, any RS232C serial port will enable you to operate the Mouse Systems mouse.

In actuality, however, I spent most of a day coaxing the PC Mouse into operation, because some serial ports will *not* work with Mouse Systems' driver software, and I happened

PROCEDURE FONTGR (FONT NUMBER : INTEGER); Selects one of four available fonts for screen text display, including two proportional-spaced fonts

FUNCTION LENGTHGR (CONST TEXTSTRING : LSTRING) : INTEGER; Returns pixel half-length of TEXTSTRING for centering tasks

PROCEDURE WRITEGR (CONST TEXTSTRING : LSTRING); Writes TEXTSTRING to screen at current graphics cursor position

FUNCTION QUERYCOMM : INTEGER; Returns indication as to whether COM1, COM2, or both are installed in system

PROCEDURE INITMOUSE (COMM PORT : INTEGER); Initializes the specified comm port for mouse operation

PROCEDURE READMOUSE (VAR X,Y,FLAGS : INTEGER); Returns mouse *X,Y* position and button flags' status

PROCEDURE STOPMOUSE; Disables mouse interrupt service processing

PROCEDURE GETCMDLINE (VAR CMDSTRING: LSTRING); Returns the "command tail" from program invocation line

PROCEDURE PUSHGRAFIX; Saves current graphics environment on the system stack

PROCEDURE POPGRAFIX; Restores the graphics environment from the system stack

PROCEDURE USERMOUSE (VAR MCB : MCBTYPE); Initializes and enables the user-written mouse interrupt service routine

PROCEDURE MOUSESVC (DELTAX, DELTAY, SW : INTEGER; VAR MCB : MCBTYPE); User-written mouse service routine for moving cursor, etc.

PROCEDURE USERTIMER (TIMECOUNT : INTEGER; VAR CCB : CCBTYPE); Initializes and enables user-written mouse interrupt service routine

PROCEDURE STOPTIMER; Disables the timer interrupt service routine

PROCEDURE TIMERSVC (VAR CCB : CCBTYPE); User-written timer interrupt service routine

PROCEDURE SETMOUSE (ID : INTEGER; CONST PARM STRING : LSTRING); Allows the user application program to dynamically redefine the MOUSESYS system resident mouse driver control functions

to have one such port.

My PC contains two quarter-meg multifunction RAM cards. One is a standard Quadboard, and the other is an Ultraboard by Advanced Data Technology. Both have RS232C serial ports that have always worked well with my modem and printer. When I plugged the mouse into the Ultraboard, the PC refused to acknowledge that the mouse was there. I put an RS232C "breakout box" between mouse and serial port and saw that data was coming out of the mouse. I was even able to read data from the mouse through small programs written in BASIC. However, the driver program sitting atop DOS in RAM was unable to pick up any data from the serial port by itself.

I called Mouse Systems in California, and although it was a Saturday, somebody was there; I soon was able to discuss the problem with a knowledgeable technician. Some serial boards were troublemakers, he said. If a manufacturer "fudges" the IBM async card spec a little to save money or circuit board space, the mouse software might have trouble.

And so it was. Bottom line, after an hour or so of powering-down and having the dip switches on the two multifunction boards, was this: with Ultraboard as COM1:, mouse would not work to either COM1: or COM2:. With Quadboard as COM1:, mouse worked to COM1: but not COM2.

Standard PC Mouse Software

Mouse Systems' approach to software is pragmatic and more immediately useful than Microsoft's. The company chose six of the most popular IBM PC application programs and wrote drivers to link its mouse to those programs. So unlike buying the Microsoft Mouse, if you buy a Mouse Systems mouse you can be mousing around in WordStar or Lotus 1-2-3 inside of fifteen minutes.

Application software packages presently supported by Mouse Systems are Lotus 1-2-3, VisiCalc, Volkswriter, Multiplan, IBM Personal Editor, and WordStar. The drivers work under DOS 1.1 or DOS 2.0. The size of each driver is different, depending on how it works with its particular application, but the drivers average about 7K each.

Installing PC Mouse support for any of the above packages involves copying a number of files from the PC Mouse disk to a working disk containing the desired application. A separate driver file exists for each application. A batch file first loads the driver as an "exit, remain resident" DOS extension, and then it invokes the application program.

The drivers do not "patch" the application programs. They work with the keyboard buffer instead, inserting commands into the keyboard character stream. Commands can still be entered through the keyboard, in parallel with the driver.

The niftiest aspect to the PC Mouse drivers are the "pop-up" command menus that appear when one of the three mouse buttons is pressed from within a supported application. A rectangular window appears in reverse video with a number of commands inside it. With the mouse you move a black bar up and down, highlighting a command, until you press the button again. The command that was highlighted when you press the

button the second time will be executed. The menu then disappears, leaving the screen text beneath it completely unaffected.

There are many command menus for each application program. These menus are linked; the user can chain from one to another by selecting a command with the "bounce bar."

The PC Mouse user guide contains "road maps" outlining which menus lead to which other menus. It sounds complicated, but it is easier to do than to describe. With a little study and some practice it becomes amazingly easy. I was zapping around the WordStar menu map like a pro inside of an hour.

Since each application program has its own driver, you have to remember to reboot the computer when you want to change from one application to another. If you simply exit WordStar and load 1-2-3 without rebooting, the WordStar mouse driver remains active in RAM; when you press the mouse buttons the pop-up menus will obediently show you the WordStar commands and send WordStar keystroke sequences if you select commands. What 1-2-3's response will be is uncertain, but it probably won't be too orderly.

My overall response to this feature of the PC Mouse is tremendously positive. If you own one or more of the supported applications, you're in business in a hurry. Between my own system and a friend's Compaq, we tested all applications but Volkswriter. The Microsoft Mouse now supports only Microsoft's application programs or programs you write yourself. PC Mouse is potentially much more useful than Microsoft Mouse.

PC MOUSE'S MICROSOFT EMULATOR

At this writing, Mouse Systems has just released its emulator for the Microsoft Mouse. The emulator supports all Microsoft's mouse driver functions in a way that is transparent to the application using the mouse. The application doesn't know (and

LEGACY.

...there'll be a Legacy to thank. The ***Legacy Series*** *offers the IBM PCjr all that it wasn't supposed to have. Now junior can have the power, storage, expansion and performance it needs to rival the PC and PC XT — all compliments of Legacy Technologies, Ltd.*

LEGACY I

Base system (standard)
- *Expansion cabinet*
- *Power supply*
- *4 slot expansion bus*
- *LED status display panel*
- *Cabling to PCjr I/0 bus*

LEGACY II

- *Base system, LTL floppy disk controller and Qume half-height floppy disk drive (the same drive that is shipped with the PCjr).*

LEGACY III

- *Base system, LTL hard disk controller and hard disk drive. (Available in 2nd quarter.)*

OPTIONAL BOARDS

- *Floppy disk controller — to control two PCjr floppy diskette drives*
- ***Legacy "EXP"*** *— 64K to 256K dynamic memory expansion capacity, two cards allow you to reach DOS maximum of 640K*
- ***Legacy "CPS"*** *— includes a clock/calendar with battery backup, parallel and serial ports*

"The first and finest add-on products for the PCjr." Dealer inquiries welcome.

CIRCLE NO. 178 ON READER SERVICE CARD

PCjr and Legacy...
one prodigy after another.

LEGACY TECHNOLOGIES, LTD.

Suite 100 • 1414 "O" Street
Lincoln, NE 68508
(402) 475-PCJR

doesn't care) which manufacturer's mouse is on line. Any application that runs the Microsoft Mouse can now run the PC Mouse instead.

I tested the emulator on Microsoft's Multi-Tool Notepad and on a graphics generator I wrote myself using Microsoft's mouse function calls. The Notepad worked fine. My own package worked well with the PC Mouse masquerading as a Microsoft Mouse until I moved into high resolution. Then the mouse driver stopped working. The people at Mouse Systems say they know what is wrong and will fix it soon.

THE OPTIONAL MOUSE WINDOW PACKAGE

The PC Mouse package contains no graphics support other than that obtained by emulating a Microsoft Mouse. Instead, Mouse Systems has gathered all its graphics support into an optional $75 package called Mouse Window. It is one high-powered piece of artillery.

Mouse Window is a linkable library of routines written in assembly language, plus several demos and a reasonable amount of documentation. Together these routines allow compiled languages to do most of what IBM's BASICA 2.0 interpreter does for graphics. Support is currently included for IBM's Color Graphics Adapter and the Hercules hi-res graphics card. Support for others is planned.

> *The PC Mouse package contains no graphics support; all its graphics support is in an optional $75 package called Mouse Window. It is one high-powered piece of artillery.*

The routines in Mouse Window are intended to be called from Microsoft Pascal or from the IBM Macro Assembler. Mouse Systems says interface information for Lattice and Microsoft C (actually the same compiler) is available upon request. Why they didn't just tuck it into the book is something of a mystery. Here and there in the documentation are promised other bits of information that are not present but should be.

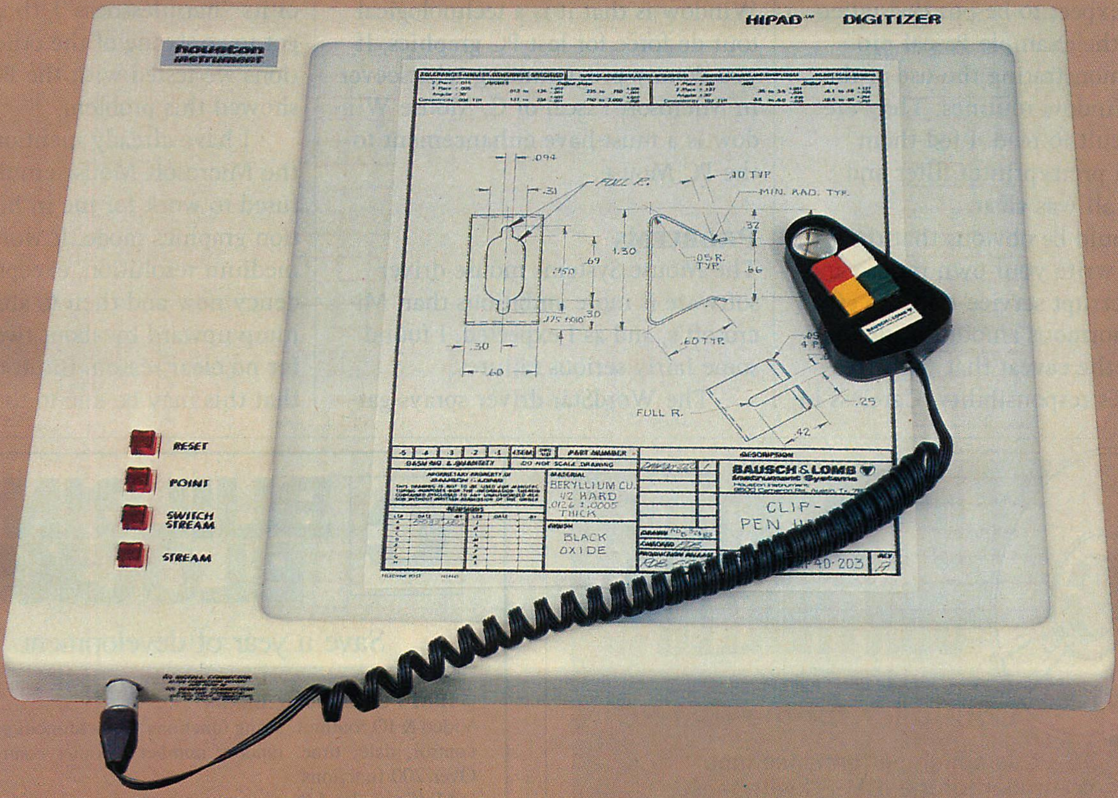
I've given the entire (and rather lengthy) list of Mouse Window routines in table 2 to give you a flavor of just how powerful a tool kit this is. Calling conventions are given in the documentation for Microsoft Pascal and IBM Macro Assembler. Methods for calling the routines from other compiled languages can probably be doped out by working backwards from the IBM Macro Assembler calling conventions.

Although I could not try as many of these routines as I would

Finally, Selecting The Right Digitizer Is Easy. Here's Why:

The HIPAD™ DT-114 is the most versatile desk-top digitizer on the market.

. . . And It's Made In The U.S.A., Certified By UL and CSA, And Meets FCC Class B Requirements



No Other Manufacturer Combines The Following Features:

- A full 11 x 11″ digitizing area.
- Graphics generation, or selection of 'library figures' from a menu.
- Translucent drawing area for tracing complex originals or radiographs.
- Meets FCC Class B requirements, to make life easier for you and your neighbors.
- Resolution of 0.010″ or 0.005″.
- Coordinate locations accurate to ±0.015″.
- Four-button cursor for selecting data-entry options.
- Your choice of outputs—all built in—to match either serial or parallel requirements . . .
  . . . if serial, both RS-232-C and TTL levels are provided on the standard DB-25P connector.
  . . . if parallel, both binary and BCD protocols are also provided on the standard DB-25P connector.
- Output rates of up to 100 coordinate pairs per second to meet the requirements of your computer.
- Coordinate pairs sent to your computer singly or in streams; construct figures dot-to-dot or freehand.
- Fixed or relocatable origins.
- Scaling in either English or Metric dimensions.
- UL and CSA listed for safe, reliable operation.
- And lastly, no stray magnetic fields to erase precious disks.

All Of The Features Listed Above Are Standard. Even The Power Supply Is Included. Now You Should Ask Us About Optional Features, And Other Versions Of The HIPAD.

For the name, address and phone number of your nearest distributor, write Houston Instrument, 8500 Cameron Rd., Austin, Texas 78753. Phone 512-835-0900, or 800-531-5205 if outside Texas. In Europe contact Bausch & Lomb Belgium NV., Rochesterlaan 6, 8240 Gistel, Belgium. Tel. 059-27-74-45, tlx 846-81339.

BAUSCH & LOMB
houston instrument *division*

HIPAD is a trademark of Houston Instrument 4P010

CIRCLE NO. 216 ON READER SERVICE CARD

have liked, what they represented to me, mouse or no mouse, was the possibility of doing real graphics work without resorting to interpreted BASIC. The purely graphics routines (those not referencing the mouse) should have been included as a library in IBM/Microsoft Pascal.

Mouse Window's documentation is a touch sparse but usable by canny programmers. Expect to be puzzled if you have no background in IBM/Microsoft Pascal. Even if you know Pascal well, expect to be puzzled when you read the example Pascal programs demonstrating the use of the Mouse Window routines. They are very difficult to read. I fed them through a prettyprinter filter and suddenly all was clear.

It should be obvious that the ability to write your own mouse and timer interrupt service routines gives you an enormous amount of flexibility—with the caveat that with freedom comes responsibility. Careless interrupt service routines are the easiest way I know to crash your system in grand style. Tyros should use the routines from the example programs until they know what they're doing. On the plus side, the proportionally spaced character fonts for graphics mode are truly gorgeous and give you the ability to fit more than 80 characters on a single line, though how many more is dependent upon which characters you use a lot.

My overall evaluation of Mouse Window is that it is a technological tour de force for fast PC graphics. If you do any programming whatsoever in Microsoft Pascal or C, Mouse Window is a must-have enhancement to the PC Mouse.

PROBLEMS

The Mouse Systems mouse driver software is more ambitious than Microsoft's, and as I expected, I found some fairly serious failures.

The WordStar driver sprays garbage characters at the screen when the mouse is moved too quickly over the pad. These characters do *not* become a part of the document, fortunately, and vanish forever when you scroll them off the top or bottom of the screen. A friend and well-known writer who tried the PC Mouse with WordStar was nearly panicked by their appearance, however, since she is strictly a computer *user* and knows only what she sees on the screen. No matter what it takes (and regardless of its "harmlessness"), this quirk has got to go. None of the other applications we tested with the PC Mouse showed this problem.

I have already mentioned that the Microsoft Mouse emulator refused to work for me in high-resolution graphics mode. It worked fine in medium resolution, except for a tendency now and then to abruptly jump upward by about twenty pixels for no clear reason. Intuition suggests that this may be due to the failure of

the mouse to correctly encode rapid movement into its rather slow (1200 baud) serial data stream.

If I were forced to choose one mouse over the other, I would have to choose Mouse Systems and swallow a couple of bugs. The fact that the PC Mouse can be used immediately with copies of popular major application programs gives it a titanic advantage.

Mouse Systems is aware of these problems and claims to be working on solutions to them.

Sadly, I did not have enough time with Mouse Window to determine its reliability. It was easily my favorite aspect of the Mouse Systems product line.

PREFERENCES

If I were forced to choose one mouse over the other, I would have to choose Mouse Systems and swallow a couple of bugs. The fact that the PC Mouse can be used immediately with copies of popular major application programs gives it a titanic advantage.

Microsoft's documentation is more organized, but Microsoft's mouse doesn't accomplish as much. I'd bank on the PC Mouse hardware to be more reliable than anything dependent on mechanical parts. Furthermore, the PC Mouse's switches never seem to "miss," as Microsoft's do with some regularity.

One objection to mice in general and optical mice in particular is a purely pragmatic one: busy people have cluttered desks. There is rarely much free space for mice to roll around on. With the Microsoft Mouse, I soon found the answer to that problem: roll the mouse around on top of the junk. Papers, binders, listings, diskette jackets, even a saran-wrapped ham sandwich afforded sufficient friction for Microsoft's mouse. I have been known to run the mouse up and down my leg when the pile of debris on my desk is too high.

To use the PC Mouse you must allocate a 9" × 12" region on your desk to accommodate the metal encoder pad. There is no substitute for the pad and a place to put it.

Microsoft did not satisfactorily answer my inquiries about the problems in the mouse; Mouse Systems was very cooperative and helpful.

For users who are of a technical bent and like to do graphics programming, the PC Mouse (with Mouse Window) would have to be the choice. By accepting Microsoft's mouse interface as a *de facto* standard and supporting it, Mouse Systems has given its product an undeniable advantage. Given a little more time to shake some bugs off of its product, Mouse Systems could turn its product into the Mouse that Roared.

Jeff Duntemann is a programmer for Xerox Corporation in Rochester, New York. His first book, Pascal from Square One, *will be published next January.*

The Microsoft Mouse
Microsoft Corporation
107000 Northrup Way
Bellevue, WA 98004
800-426-9400
$195
CIRCLE 494 ON READER SERVICE CARD

PC Mouse
Mouse Systems Corporation
2336H Walsh Avenue
Santa Clara, CA 95051
408-988-0211
$295
$75 for Mouse Window
CIRCLE 495 ON READER SERVICE CARD

Die Thermalgebiete gehören zu den größten Reichtümern Islands.

PROOFWRITER™

Word Processor, Program Editor, and Spelling Checker

PROOFWRITER™ is a comprehensive, easy-to-learn, menu-driven Word Processing and Program Editing package designed for Scientific, Multilingual and Business Applications.

The Main Features of ProofWriter™ are:

- Full Screen Menu-Driven Editor
- Over 60 Single Keystroke commands
- Word Processing in Color
- On-Screen Tutorial Introduction
- Spelling Checker
- Mail Merge Capabilities
- "Help" commands
- Easy "cut and paste" capabilities
- Spelling Errors Displayed in Reverse Video
- Foreign Language and Scientific Symbols inserted with single keystroke
- Underlining, emphasized text, superscripts and subscripts displayed during editing
- Equation mode for easy entry of Scientific Equations
- Constructs and prints any Foreign Language or Scientific Symbol on any matrix printer

DEMONSTRATION DISK WITH INTERACTIVE TUTORIAL AVAILABLE FOR $5.00

$$\int_0^\infty d\Omega\, A_{00}(\Omega)\, e^{i\Omega t} A_0(r,\Omega) = 2\pi \sum_\alpha G(\alpha)$$

- Any symbol not included in the IBM PC character set can be displayed with optional character PROM $125
- Tabs, Margins, and Text reformatted on the screen
- Inserts phrases or blocks of text with single keystroke
- Columns of data can be moved
- Many unique printing options: underlining on any printer, superscripts, subscripts, software generated shadow and bold type, right/left justification, centering, *italics;* enhanced, enlarged or condensed print; automatic numbering of footnotes, endnotes; headers, footers, and numerous page numbering options
- Will work with any printer: matrix, thimble or daisy wheel
- Text completely DOS compatible
- Compatible with program compilers

Only $250

Manual Only — $30 *Visa or MC accepted*

Order direct or thru your Computer Dealer

IBM XT or PC with 2 Disk Drives; 192 KB Recommended

Image Processing Systems **608/233-5033**

6409 Appalachian Way • P.O. Box 5016 • Madison, WI 53705

$$u \frac{\partial}{\partial x} [\delta] = k_F [\delta][I] - k_R [\sigma][I]^*$$

$$z_p = \frac{\iiint u^2 du}{[2 + 4(-1)]^3}$$

$$\bar{x} = \sum_j e_{ij} x_i$$

What Do BASIC Error Messages Really Mean?

TECH NOTEBOOK 16

A quiz to test your knowledge of confusing BASIC error messages

NELSON FORD

If you program in BASIC, you have probably at least once spent half an hour trying to find the cause of an error message, only to realize finally that the error had little logical connection to the message. Obviously, error messages can mean many things. The first thing you should do when you receive such a message is look in the manual to see what trick meanings it could possibly have. If you think you understand these messages, try this quiz:

1. Which of the following program statements is most likely to result in the message "BAD FILE NUMBER"?

a. OPEN "EXPERIMNT.DTA" FOR INPUT AS #1
b. OPEN "TEST.DTA" FOR OUTPUT AS #7
c. OPEN A LOCK WITH A #2 WOOD FILE

2. Which of the following will result in "TOO MANY FILES"?

a. SAVE "B:PROGRAM" with the switches set for one drive
b. OPEN "TEST.DTA" FOR OUTPUT AS #7
c. Adding an extra zero to an order for 10 file cabinets

3. Which of the following will result in "LINE BUFFER OVERFLOW"?

a. Neither of the below
b. Entering 500 characters in a program line
c. Neither of the above

4. The statement OPEN "A:A:FILENAME" AS 1 LEN=20 will result in which of the following error messages?

a. FILE NOT FOUND
b. BAD FILE NAME
c. SYNTAX ERROR
d. ILLEGAL FUNCTION CALL
e. anything else that makes sense, or
f. TOO MANY FILES

5. The statement OPEN "PTR1:" FOR OUTPUT AS #1 (an attempt to open the printer for output—the proper syntax is OPEN "LPT1:") will result in which of the following error messages?

a. FILE NOT FOUND
b. BAD FILE NAME
c. BAD FILE NUMBER
d. SYNTAX ERROR
e. anything else that makes sense, or
f. TOO MANY FILES

6. Which of the following pieces of code will result in the message "INPUT PAST END"?

a.
```
10 OPEN "TEST.DTA" FOR
   INPUT AS #1
20 END
30 INPUT #1, A$
```
b.
```
10 OPEN "TEST.DTA" FOR
   OUTPUT AS #1
20 INPUT #1, A$
30 IF NOT EOF (1) THEN
   GOTO 20
```
c. ..-.-... --- ..-- .-. -- .. .. ---.

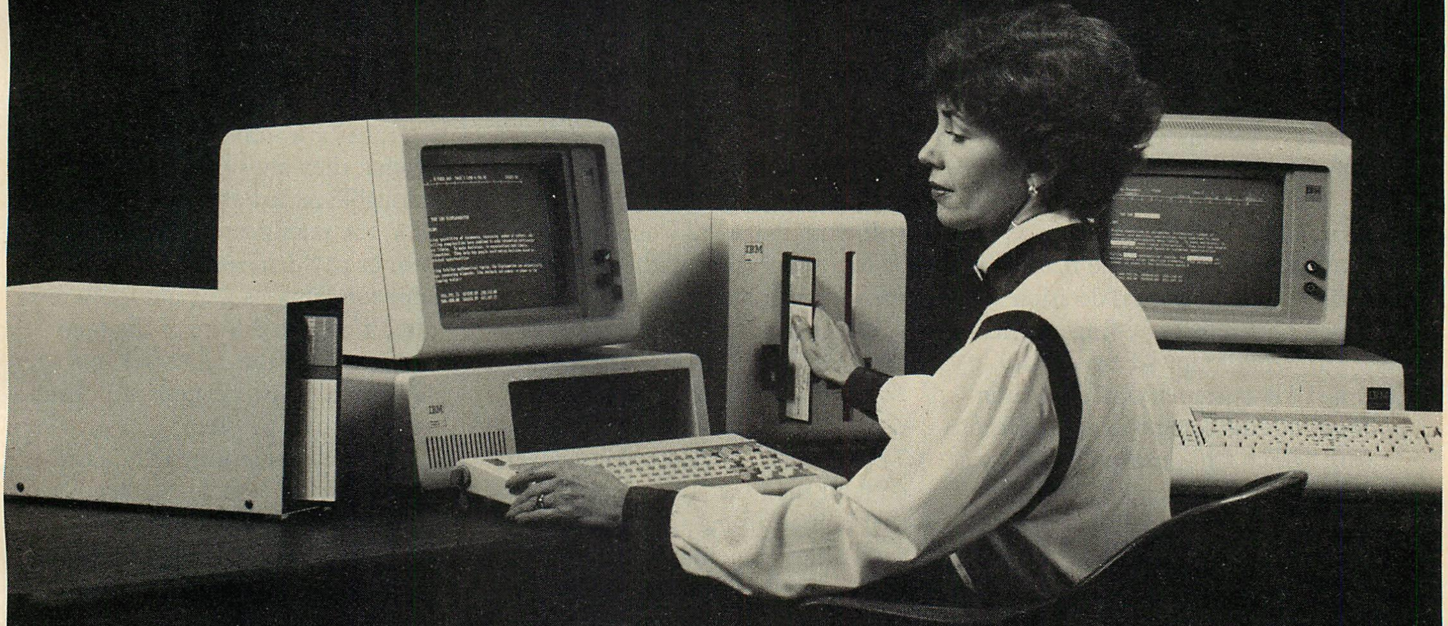
Answers:

1. *A)—Opening a file to buffer #7 will give you an error message only if you did not open enough file buffers when entering BASIC. Answer A will always generate the message "BAD FILE NUMBER" because the file name has too many letters in it. Makes sense, doesn't it?*
2. *A)—Accessing a drive beyond the motherboard switch settings will result in this message. Surely you didn't think that specifying a non-existent drive would result in an error message like DEVICE UNAVAILABLE.*
3. *?)—As so often happens in real life, there is no correct answer for this one. I have never been able to generate a LINE BUFFER OVERFLOW error by entering too many characters in a program line. BASIC truncates the excess without warning.*
4. *F)—"TOO MANY FILES" may not seem like a logical error message for trying to use a bad file name, but this is not a test on logic.*
5. *C)—"BAD FILE NUMBER". Another bad file name, but a different error message. This one doesn't make any more sense than "TOO MANY FILES."*
6. *B)—Trying to INPUT from a file opened for OUTPUT results in an "INPUT PAST END." This is not the worst of these misleading messages, but it will still cause you to waste time trying to figure why the EOF function is not working.*

Nelson Ford is Director of Reporting and Forecasting Systems for Daniel Industries, Inc. and past president of the Houston Area League of PC Users. He is the author of Business Graphics for the IBM PC *and of the disk file catalog program, DISKCAT.*

THE FILE CONNECTION

8″ DISKETTE SYSTEM FOR THE IBM PC



Our "FILE CONNECTION" programs provide 8″ diskette file exchange between the IBM PC and most Micro-Mini-Main Frame computer systems.

Our "WORD CONNECTION" programs provide 8″ diskette text document exchange between the IBM PC and many word processing systems.

Contact us for information about the hundreds of 5¼″ and 8″ diskette formats and systems which we currently support.

FLAGSTAFF ENGINEERING / 2820 West Darleen / Flagstaff, AZ 86001
Telephone 602-774-5188 / Telex 705609 FLAG-ENG-UD

Business Non-Advice

A legal perspective on some common business problems

MAX STUL OPPENHEIMER

NON-ADVICE

Lawyers play many roles—courtroom advocates, negotiators, legal advisers, goats—but most of them draw the line at playing the role of business adviser. Deep down inside, I might have reservations about introducing a computer called "Eve" and advertising it as having "all the features of Adam at 60 percent of the price," but I would probably tell the manufacturer "that's a business decision" and hope I hadn't left any fingerprints.

In what follows I have abandoned my customary practice in order to discuss some business decisions with which I have seen clients wrestle and which a fledgling entrepreneur will face sooner or later. Remember my lack of expertise in the field.

Do I need a written contract? The law is that, with certain statutory exceptions (generally, contracts involving real estate, contracts for the sale of goods in excess of $500 and long-term contracts—check your state law), contracts need not be in writing. If two people agree, with the apparent intention of being bound by their agreement, and each receives something of value, they have a contract. A written contract helps define that agreement, in case anyone has trouble remembering.

A written contract takes time, though, at least partly because most people take more time to focus on the agreement if it is to be embodied in a written contract. Often, potential problems do not surface until the parties try to write down exactly what they are agreeing to; it is usually better to realize the problem early. On the other hand, in some circumstances, it might be better to defer the problem until after the business arrangement is humming and everyone has a stake in keeping it going.

No contract is perfect; no contract will cover every contingency. Let me qualify that: maybe a perfect contract can be written, but I can't do it. The goal of a contract should be to get the parties in agreement. If that goal is accomplished, the piece of paper can be thrown away.

Can I write the contract? Yes, but your approach would be different from a lawyer's. You'd probably focus on how the arrangement will work; a lawyer would probably be more concerned with what happens when it doesn't work. Would you do a major tune-up on your own car? Does your car have fuel injection?

Can I copy a contract a friend used once and change the names? That is more dangerous, but lawyers do it all the time. The problem is that no two arrangements are ever exactly alike. Sometimes the differences are irrelevant, but sometimes they are not. Boilerplate (legal shorthand for "the lawyer who originally wrote this clause was missing in action in the Norman invasion of England and his notes seem to have been destroyed when the plague swept Europe, but we keep using the clause in all of our documents just in case someone remembers what it is supposed to mean, and we certainly hope it is to our benefit) can be particularly dangerous, because it usually does have a function. Would you rebuild your car's engine? Would you rebuild an airplane engine? Would you fly in the plane afterwards?

Can we use a standard form contract? It depends—are you entering a standard enough agreement? Keep in mind that a contract's being printed on long paper in small type doesn't mean you can't change it. Usually, you should.

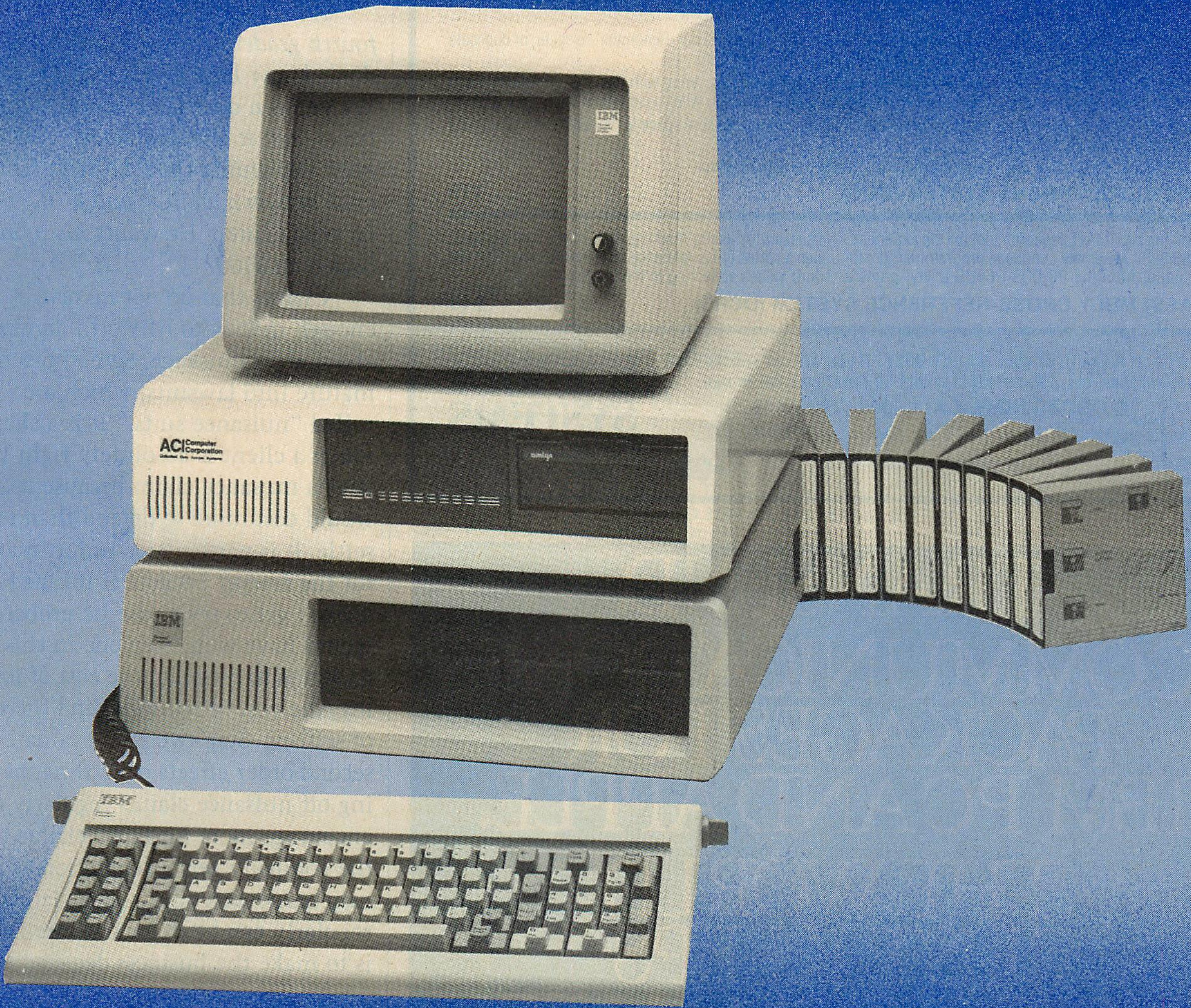
Our customers are complaining. Something in the hardware or software isn't working. It may be because we ran short of 8088 chips and used 8085's instead on a few units or because it was late when we wrote the output subroutine. Should we recall the units, issue a warning to customers who haven't noticed the problem yet, or tell our salesman to tout our new random-response function?

In the absence of a written disclaimer of warranty, the law generally implies a warranty that goods will do what they are supposed to do for a reasonable period of time. If they don't, the manufacturer must fix them or pay damages. He may also be liable for damages that naturally resulted from the failure of the product. Of course, this all comes into play only if someone finds out about the company's problem.

There is something to be said for not looking for trouble. Some customer may have just fired his programmer because he couldn't get consistent results. It may never dawn on him where the problem really is. If it does, though, he will probably be angry if he has spent a small fortune trying to solve a problem that you could have solved cheaply. How important is your company's reputation? Just how much damage might the malfunction do? How costly will repair be if you do it? If your customer has someone else do it? How many of these little gems did you sell? Who else besides you knows where the real problem with the equipment lies?

Our new machine seems to have a problem. Every once in a while it gets confused. We've fired our programmer, but our new programmer can't make the thing behave either. The salesman says not to worry, the manufacturer has looked at the machine and found nothing wrong with it. A local fourth grader was in yesterday watching the service rep and

and...
the backup is free!



With the ACI CEO unit added to your IBM or IBM-compatible computer, you get: high-speed mass storage one hundred times more reliable than tape, backup and restore ten thousand times faster than tape, unlimited off-line capacity, and . . . the backup is free!

Having an ACI CEO backup system means you'll never lose valuable data to user error or computer breakdown again.

Call now for more information on ACI's CEO randomfile backup system.

ACI Computer Corporation
Unlimited Data Access Systems

Dealer applications being accepted.

CIRCLE NO. 162 ON READER SERVICE CARD

4450 Enterprise, Suite 101 • California 94538 • (415) 656-9880 • Telex: 171618 • TWX: 910 3382290

asked how they managed to use an 8085 in place of an 8088—I didn't understand the answer.

See the discussion above. Also see an independent repairman to be sure there is a design problem. Then total up your damages and decide whether you want the machine repaired or whether you'd prefer a refund.

A customer was in today with a fourth grader in tow, claiming we don't know how to build computers. We've been over his machine a dozen times, and it works perfectly. We keep explaining that there is a difference between an "O" and a "0," but he won't listen. He wants his money back—all $100.

Claims that do not involve enough money to be worth arguing about are a nuisance. Sometimes they mature into lawsuits, which are called "nuisance suits." It is a shame when a client is absolutely right but pays a nuisance claim because it would cost more to defend than to settle. It is particularly unfortunate for the lawyer because it means he doesn't get to try a case he probably would have won. Step one in this situation is to estimate the cost of losing, the cost of winning, and the cost of settling. Step two is to evaluate the second-order effects of settling: paying off nuisance claims regularly can encourage more nuisance claims if the company gets a reputation as an easy mark; defending such claims regularly can be expensive. Step three is to make the business decision.

Max Stul Oppenheimer has an engineering degree from Princeton and a law degree from Harvard. He is a partner at Venable, Baetjer, and Howard in Baltimore.

ORYX SYSTEMS, INC.

CRAFTSMEN OF THE NEW TECHNOLOGY.
BEST PRICES WITH QUALITY SUPPORT* ON SOFTWARE AND HARDWARE.

SOFTWARE

Alpha Software
Data Base Manager II . $ 179
Mailing List 72
Executive Package..... 105

Autodesk
Autocad.............$ Call

Bullish Investment
Net Worth ....$ Call

Central Point
Copy II PC ....$ 34

Compuview
Vedit 8086 CP/M 86 ...$ 160
Vedit MS-DOS 130

d-BASE II CORNER

Ashton-Tate
d-Base II ....$ Call
Bottom Line Strategist 259
FPL 410
Friday............ 179

Human Soft
dB Plus...........$ 90

Fox & Geller
Quick Code ......$ 175
D Util 58

Sensible
D-Programmer ....$ Call

Software Banc
d-Base II User's Guide:
w/ d-Base II Purchase ....$ 15
w/o d-Base II Purchase 20

Anderson-Bell
Abstat ....$ Call

Tylog Systems
d-Base Window ...$ 159
d-Base Door ......$ Call

Decision Resources
Chartmaster ....$ 289
Signmaster ....$ Call

Digital Research
Concurrent CP/M 86 ...$ 225
CP/M-86 39
Cobol 86............ 499
Pascal MT + 86 CP/M 86 w/SPP 375
SPP 86 130
C Basic 86........... 135
PL/I 86 499

Dow Jones
Market Analyzer ......$ 245

Eco-Soft
Microstat ....$ Call

Emerging Technologies
Edix (editor) ....$ 139
Wordix (formatter) 139

Graphic Software
Super Chartman II .....$ 299
Super Chartman IV 199

Financier Products. $ Call

FYI Products ....$ Call

Harvard Associates
PC Logo..............$ 129

Harvard Software
Project Mgr. ....$ Call

Hayes
Smartcom II ....$ 89

Innovative Software
T.I.M. IV.............$ Call
Fast Graphs .... Call

Lexisoft
Spellbinder ....$ Call

Lifetree Software
Volkswriter Deluxe ....$ 179

Living Videotext
Think Tank (256K)......$ Call

Microfocus
Cobol Level II $1,250

Micro Pro
WordStar ....$ Call
Infostar ....$ Call
Pak (WS/MM/SS/Index) $ Call

Microrim
R-base ....$ Call

Microsoft
MuLisp/MuStar........$ 155
MuMath/MuSimp...... 179
Basic Comp (DOS) 389
Pascal Comp (DOS) ... 249
C Compiler (DOS)..... 329
Basic Compiler (DOS).. 249
Multiplan 165

Microstuf
Cross Talk ....$ 109

Northwest Analytical
Stat Pak ....$ 365

Oasis
The Word Plus ....$ 103
Punc & Style 90

Open Systems
Accounting Pkgs. .....$ Call

Pearlsoft
Personal Pearl Acct'g (GL, AR, AP, P, INV) ..$ 650

SPI
Open Access ....$ Call

Software Arts
TK! Solver ....$ 189
Application ea. ....$ Call

Software Publishing
PFS: Report ....$ 79
PFS: File.............. 89
PFS: Graph 89
PFS: Write............ 89

Softword Systems Inc.
Multimate (WP) ....$ Call

Sorcim
Supercalc III ....$ 249

Supersoft
C Compiler - 8086 ....$ 350
Basic Compiler........ 225
Fortran IV PC/DOS or 8086 299
8087 Support 40
Diagnostics II 89
Scratch Pad 187

Systems Plus
Landlord (prop mgmt) .$ 375
Runtime Basic (req'd for above) 45

Woolf Software Systems
Move It (PC to CP/M and PC to PC) ......$ 109

... and many more.

BOARDS

AST Research
ComboPlus 64K (Cl/Cal, Ser & Par, I/F, 256K cap.) ....$ 279
MegaPlus 64K, (Cl/Cal, Ser Port, 512K cap. w/Megapak) ....$ 269
Extra ports for Megaplus and I/O Plus II (Game, Par & Ser).....$ 40
Megapak 256K upgrade for Megaplus ....$ Call
I/O Plus II (Cl/Cal and Ser Port) ....$ 115

Quadram
Quadboard 64K, (Cl/Cal, Ser & Par Ports, Software) ....$ 279
Microfazer Stack Printer Buffer (exp. to 512K)
- Parallel/Parallel 8K ..$ 132
- Parallel/Parallel 64K . 188
- Serial/Parallel 8K 170
- Serial/Serial 8K 170

Quadlink 64K (Mem allows Apple SW to run on IBM/PC .....$ Call

Tecmar Products ...$ Call

Xedex/Microlog
Baby Blue ....$ Call

Tall Tree
512K JRAM Mem Bd ...$ 699

MONITORS

Amdek Video 300A Am ....$ 149
Amdek Color II + ... 435

NEC
JB1201 12" Gr ....$ 169
JB1260 12" Gr 119
JC1216 RGB........... 435

PGS HX12 RGB Clr ....$ Call

Sanyo 8112 12HR Gr ..$ Call

SONY
Profeel 12".......$ Call
Profeel 15".......$ Call
Profeel 25"......$ Call

Taxan
KG12-UY 12" HR Am ...$ 139
KG12N-UY 12" HR Gr . 132
RGB Vision 1 12" LR Clr $ Call
RGB Vision 3 12" HR Clr $ 459

USI 1200A 12" HR Am . $ 159

MODEMS

Hayes
Smartmodem 300 .....$ 205
Smartmodem 1200 495
Chronograph 189

Novation PC Cat w/ Crosstalk *also known as* Access 1-2-3 1200B .....$ Call

US Robotics
Auto-Dial 300/1200.... 459
Auto-Link 300/1200 410
Password ....$ Call

DISK DRIVES

CDC 1800 249

Corona
5 MB Hrd w/cntrlr..... 1,395
10 MB Hrd w/cntrlr.... 1,795

Corvus ....$ Call

Davong 10/15/20 MB..$ Call

Tall Grass............$ Call

Tandon TM-100-2 239

HALF HIGHS

Matsushita...........$ 249
Panasonic 205
Shugart 229
Teac w/brackets 259

PRINTERS

C. Itoh Starwriter F10..$1,049
C. Itoh Prowriter 8510 .$ Call
C. Itoh Prowriter 1550 .$ 599

IDS/Data Products
Prism 80 (w/4 options) $1,399
Prism 132 (w/4 options) 1,547
Microprism 479

Epson FX-80..........$ 575

GE/Genicom Printers..$ Call

Mannesmann Tally
MT 180L ....$ Call
MT 160L ....$ 585
Spirit...............$ Call

NEC 3550 ....$ Call
NEC 8023A...........$ 389

Okidata
Microline 82A.........$ Call
Microline 83A.........$ Call
Microline 92 ....$ Call

Siemens
Ink Jet Printers ....$ Call

Silver Reed
Daisy Wheel..........$ Call

Star Micronics
Gemini 10X ....$ 299
Gemini 15X ....$ Call

Teletex T1014 ....$ Call

Transtar
T-130 P/S ....$ 665
T-315 P Dot Matrix 489
T-120 P/S 475

... and much more.

DISKETTES

3M 5" DS DD Box ....$ 40
Maxell 5" DS DD MD2 Box...........$ 40
Verbatim 5" DS DD Box ....$ 35
Ultramagnetic 5" DS, DD, Bonus Box (12 diskettes) ....$ 35
(3 Boxes Minimum)

PLOTTERS

Enter P100 Sweet P ....$ Call
Panasonic Digital Plotter ....$ Call
Strobe M100 Plotter (RS 232) ....$ 499

MISC.

Alpha-Delta
"MACC" surge prot. $ 69

Keytronic
WP Keyboard ....$ Call
WP Keyboard KB KB5151 Dvorak ..$ Call

Microsoft Mouse .....$ 155

Mouse Systems
PC Mouse ....$ 229

Seattle Computer
8087 Pkg. ....$ 299

Street Electronics
Echo Speech Synth. .$ 179

PEACHTREE CORNER

- PeachText 5000 Personal Productivity System. *Includes:* PeachText Word Processor, Random House Elecronic Thesaurus, Spelling Proofreader, PeachCalc Electronic Spreadsheet and List Manager.

All of the above for ....$ Call
Call for information on other Peachtree Software.

Please call for accessories and hardware not listed.

ORDER TOLL FREE — OUTSIDE WISCONSIN — 1-800-826-1589

Please:
- Wisconsin residents add 5% for sales tax.
- Add $3.50 for shipping per software and light items. For multiple and other items, call.
- Foreign — add 15% handling & shipping for prepayment. (Int'l money order.)
- Prices are subject to change without notice.
- All items subject to availability.

Store prices are strictly retail.

We Welcome:
- Visa, MasterCharge & American Express. (Add 4%.)
- Checks. (Allow 1-2 weeks for clearing.)
- COD (Add $2.00 per shipment. Cash or certified check.)

Working Hours: Monday-Friday 8:30-5:30
Saturday 10:00-2:00 • Central Time

***For technical assistance, order status and in Wisconsin, please call 715-848-1374. We will cheerfully oblige.**

ORYX SYSTEMS, INC.
CRAFTSMEN OF THE NEW TECHNOLOGY

425 First Street
P.O. Box 1961
Wausau, WI 54401

Int'l Telex: 260181
ORYX SYS WAU

CIRCLE NO. 181 ON READER SERVICE CARD

PC Mouse works wonders.

Works with Lotus 1-2-3, VisiCalc, VisiON, WordStar, Microsoft Word, Multiplan and IBM Personal Editor.

Works with existing hardware and software.

No modifications are required. PC Mouse connects to your IBM PC, PC-XT and compatible PC through the RS-232 port. Software installation is a one-time, one-line procedure.

And if you don't like our menus, you can design your own with our *Designer Pop-up*™ menu software.

Helps you work quickly.

PC Mouse is ideally suited for moving the cursor. It's fast. Eliminates using the cursor keys. Use the pop-up menu to enter a command by pressing a button on the mouse. It's simple. Nothing to memorize.

And it can work for you.

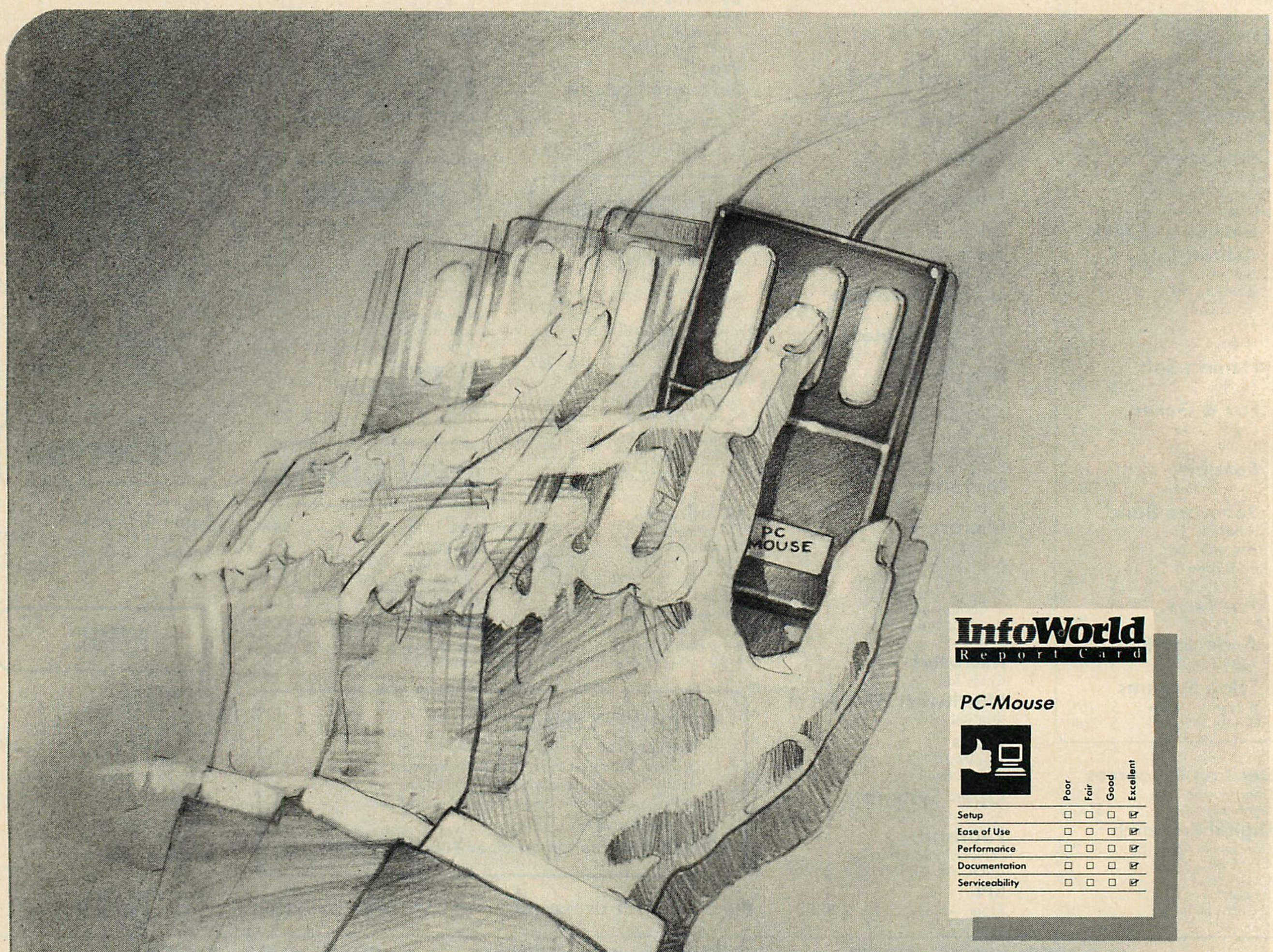
PC Mouse is so fast and so impressive it can help you sell-through compatible equipment.

PC Mouse is available from Softsel or Mouse Systems. Suggested retail is $295.00 complete. To help *you* work wonders, PC Mouse is available one-time-only from Mouse Systems for $150.00.

For more details, contact us at 2336H Walsh Avenue, Santa Clara, CA 95051. Telephone (408) 988-0211 or Telex 467848.

Pointing the way.

MOUSE SYSTEMS



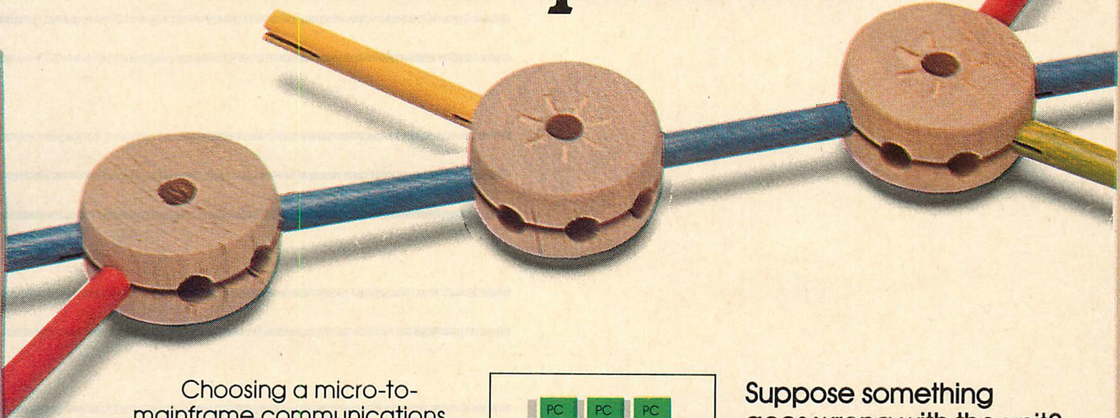
© 1983, Popular Computing, Inc., a subsidiary of CW Communications, Inc. Reprinted from InfoWorld.

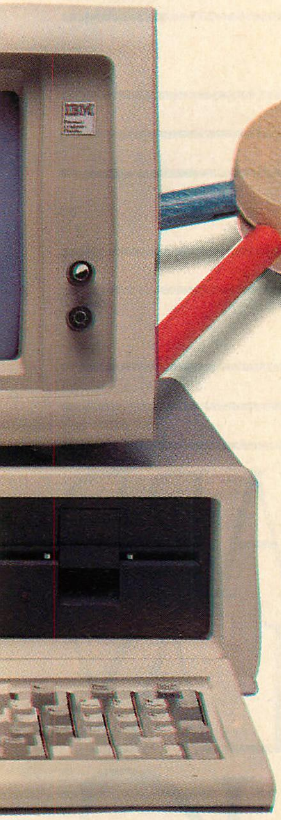
CIRCLE NO. 112 ON READER SERVICE CARD

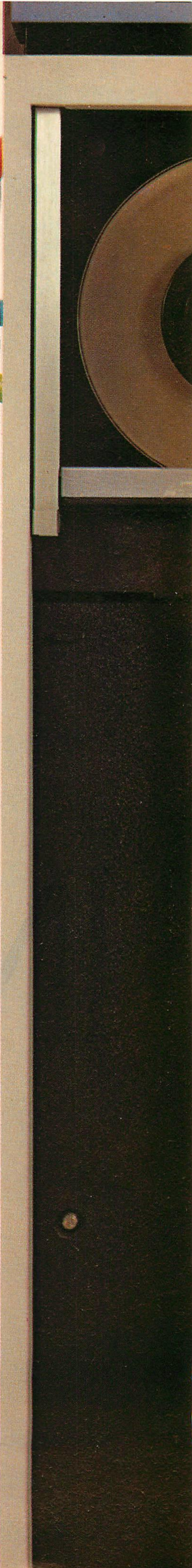
IBM, PC, XT are trademarks of IBM Corporation.
1-2-3 is a trademark of Lotus Development Corporation.
Multiplan, Microsoft Word and Microsoft are trademarks of Microsoft Corporation.
VisiCalc and VisiON are trademarks of VisiCorp.
WordStar is a trademark of MicroPro International

Micro-to-mainframe:

Before you settle for a simplistic solution, ask a few serious questions.

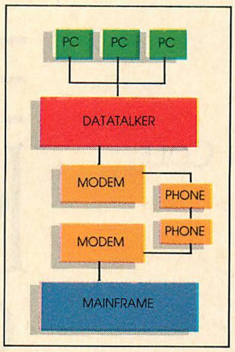






Choosing a micro-to-mainframe communications system is no game.

So before you toy around with "easy" solutions, ask some serious questions.



Will this product support full IBM Terminal Emulation?

Make sure the system can emulate remote batch and interactive IBM terminal systems.

Does the company offer a variety of products?

You should have your choice of stand-alone front-end processors, IBM PC or XT boards, or an OEM board. All ready to run on the most popular operating systems.

Can I get fast answers to my questions?

Insist on toll-free access to qualified service personnel before and *after* the sale.

What if I need a quick analysis of a problem?

Ask if the manufacturer has a Communications Test Center for product testing over public phone lines. And find out if the system has internal diagnostics.

Suppose something goes wrong with the unit?

The company should offer a 30 day money-back guarantee and a 12 month warranty that includes a free replacement unit.

What about future product development?

Make sure the products you're going to need soon will be available soon.

Who am I dealing with anyway?

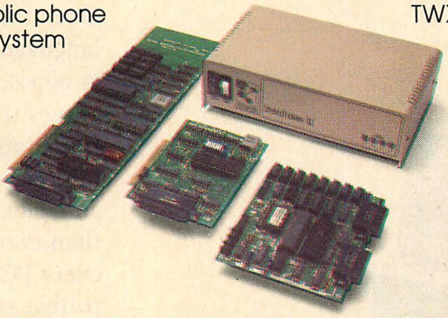
How long have they been in business? The longer, the better.

How much is all this going to cost?

Here's a point of reference: Our own DataTalker product line will give you all the capabilities and services just described for as little as $695. Complete. If you'd like to know more, contact our Marketing Support Group at 1-800-321-7785.

Providing Data Communications Solutions

Winterhalter, Incorporated
3853 Research Park Drive
P.O. Box 2180, Ann Arbor, MI 48106
313/662-2002 800-321-7785,
TWX 810-223-2423
TELEX 234-216



CIRCLE NO. 274 ON READER SERVICE CARD

DataTalker™

By Winterhalter, Incorporated

Winterhalter, Incorporated has been providing data communications solutions to major manufacturers of microcomputers and word processors since 1978. Our DataTalker front-end processors and PC boards are used by companies around the world, from start-up manufacturers to the Fortune 1000.

NASDAQ SYMBOL WNTL

JOHN COLE

KEYBOARD & DISPLAY INTERFACING

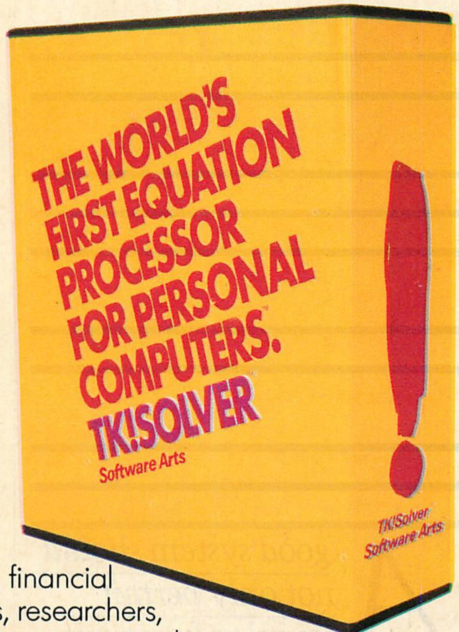
Routines for a standard interface to the PC's keyboard and display

One of the pleasant things about writing assembly language programs for the IBM PC under MS/DOS as opposed to writing them for the myriad of machines that run CP/M is that certain of the system functions are relatively standardized, particularly the interface to the display, the keyboard, and the time and date. There is only one keyboard and no more than two different types of monitor; in addition, every PC and every machine that claims to be compatible with the PC keeps a clock. This contrasts with the more than thirty terminals that interface to

John Cole is a consultant in the Dallas area, specializing in the IBM PC. He is a former instructor of computer science at Illinois Institute of Technology.

EQUATIONS PROCESSED

NO PENCIL. NO PAPER. NO MANUAL LABOR.



The TK!Solver® program will take on your toughest problems—linear, quadratic, simultaneous equations, whatever. Then stand back. Because TK!Solver turns your personal computer into a simple, yet powerful, desktop equation processor.

Whether your problem is a simple formula or a model consisting of many equations, TK!Solver can help improve your productivity. Once the equations are written, enter the known values, press the ! key, and TK!Solver gives you the answer.

Engineers, scientists, architects, financial analysts and planners, educators, researchers, and other professionals who use equations and mathematical models can work more creatively with TK!Solver.

TK!SOLVER GIVES YOU:

BACKSOLVING

If the programs you use now require you to rewrite the same equation to solve for different unknowns, TK!Solver can dramatically improve your productivity. Enter your problem once and then solve for the unknowns no matter where they are in your equation.

ITERATIVE SOLVING

If TK!Solver can't solve an equation directly, take an educated guess at the answer. Type the ! key and the TK!Solver program starts with your guess and performs repeated approximations to converge on the answer.

LIST SOLVING

Given a list of input values, TK!Solver automatically calculates the equation for every value in your list. For example, if you want to know how different interest rates will affect monthly loan payments, enter a list of interest rates and let TK!Solver calculate the payment amount for each value.

UNIT CONVERSIONS

Any type of unit conversion—Fahrenheit to Celsius, meters to feet, dollars to deutschemarks, newtons to dynes—can be made without altering your equations. Just define the numerical relationship between two units of measurement and the TK!Solver program automatically converts the variable value to the unit you specify.

TABLES AND PLOTS

Quickly generate tables and plots of your results on your screen or printer.

AVAILABLE NOW

You can run the TK!Solver program on the IBM® PC and XT and compatible machines, the Digital™ Professional™ 350, the Digital™ Rainbow™ 100, the Wang Professional Computer, Apple® //e, and on the following personal computers using MS™-DOS: TI Professional Computer, GRiD Compass Computer,™ Canon AS-100, Eagle® 1600, Toshiba T300, and the Zenith Z-100.™

SEE IT TODAY

There's more. Lot's more. But you'll have to see it to believe it. And that's easy. Bring your own equations into your nearest computer retailer and ask to see the TK!Solver program in action. The world's first equation processor for personal computers.

TK!Solver®

By Software Arts,™ creators of VisiCalc®

27 Mica Lane, Wellesley, Massachusetts 02181 617-237-4000

CIRCLE NO. 280 ON READER SERVICE CARD

probably as many machines running CP/M; some of those machines have clocks and some do not.

Despite this advantage, the PC has its limitations. Certainly between MS/DOS and the ROM BIOS, it is possible to manipulate the keyboard, display, clock, and calendar to do just about anything. However, a good system should not only permit complex functions to be done; it should also provide simple functions that are easy to use. A standard interface to the keyboard and display of the IBM PC would clearly make it easier to use the machine; the routines that follow (see listings 1-3) were written to provide such an interface.

An example of the problems posed by the PC is function call 9, which displays a string from memory. A string displayed using function call 9 will be terminated with a dollar sign. What if you're British? Have you ever tried to print a balance sheet or a payroll ledger without a dollar sign? You will quickly discover that this function provides no control over the rest of the screen—no way to position the cursor, to highlight, to scroll, or to do much else.

At the other end of the spectrum, there are several BIOS calls that let us do anything we want if we can figure out how to indicate our desires using those calls.

The keyboard entry routine, function call OB hex, is not as bad as those mentioned above. Its only problem is that it is not really compatible with function call 09 hex; thus it is difficult to redisplay whatever was just keyed in. The count is useful only if you plan on going by it rather than by the terminator, which is always the carriage return.

The time and date routines provided by function calls 2A and 2C hex respectively are fine; one would not expect much more from a microcomputer operating system. Anyone who needs these numbers in more palatable form can write some simple routines to transform them. Each routine turns out to be just about a page of code on the 8088, which is not bad.

A good system should not only permit complex functions to be done; it should also provide simple, easy-to-use functions.

One of the general requirements of the original project for which the following routines were written was that they use no data area. They were written to be included into code, rather than linked in, largely because it is somewhat difficult to convince LINK to put things where they should be. Making these routines into linkable, relocatable modules involves defining the entry points as external, defining a code segment, and telling the assembler what assumptions to make about them.

Let's take a close look at each of the routines. The S$DSPLY routine is the most complex and the most useful. It provides in a single display string control codes that allow the caller to do the following things: position the cursor, erase to the end of a line, erase to the end of the screen, cause a beep (bell character), and scroll the screen both up and down. In addition, the display can be terminated by two different characters with different effects. A carriage return terminator positions the cursor at the beginning of the next line before exiting, while an ETX (ASCII end of text) causes the cursor to remain positioned just after the last character displayed. This is handy if the next thing is a keyin, as is often the case when you are filling in a form or answering questions.

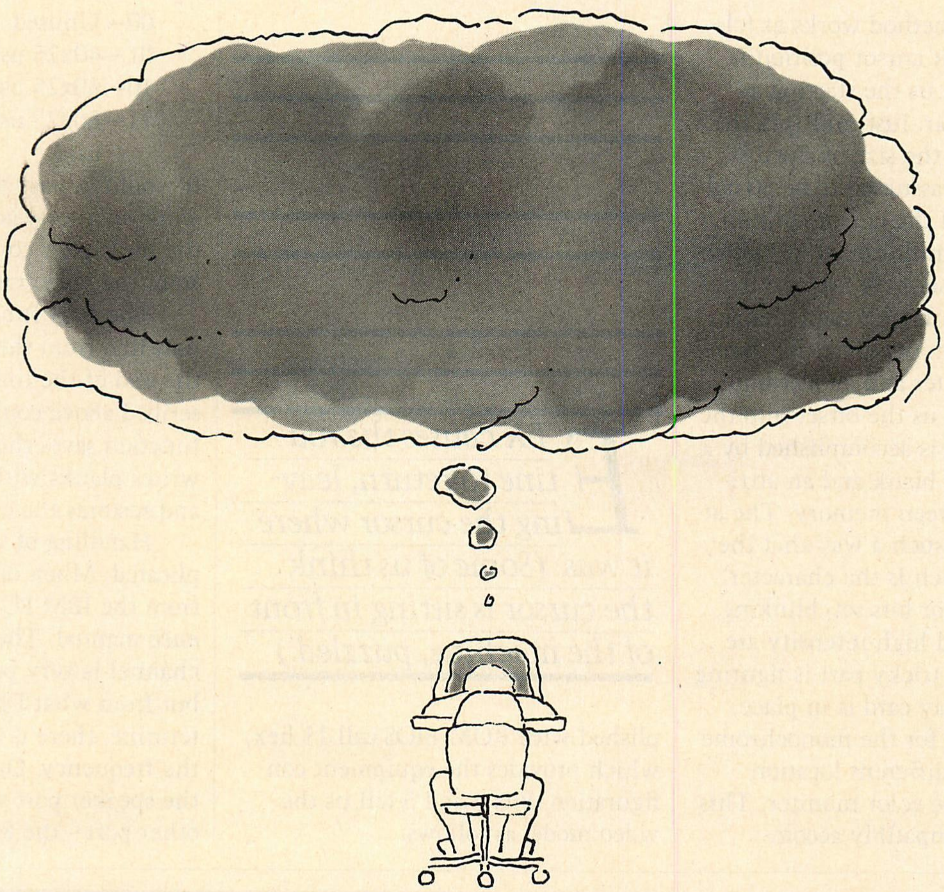
All control codes are done with equates, since what was useful for the project for which these routines were originally written may not be useful for the next project. Note also the use of the SVC macro and equates for the function call numbers. There are few more frustrating things than trying to figure out just by looking at the listing what something was intended to accomplish.

The routine uses the MS/DOS function call 2, which displays a single character, for all screen output. Otherwise it uses the ROM BIOS.

Screen scrolling could be done with a carriage return, except that we are not always assured that we are on the bottom line. Therefore this routine uses the ROM BIOS to make sure the entire screen moves.

The cursor is positioned when the routine interprets the proper code. The position follows in the order that is expected by the BIOS. Note that the screen position can contain anything, and it is the caller's responsibility to make sure it contains a valid value.

Clearing from the cursor position to the end of the screen is another routine that requires some explanation. There are clearly two ways to do this. The first, which is very slow, is to save the cursor position and then use function call 02 to display a blank in each of the remaining positions on the screen. The other way, which involves accessing the display memory directly, is much faster, although the screen does flash momentarily while it is being cleared.



The latter method works as follows: the current cursor position is retrieved to give us the starting address for the clear. Just multiply the row number by the size of the display times two (assumed to be 80 columns), then add the cursor column. The additional multiplication by two is needed because there are two bytes in the screen buffer for each character on the screen: one byte that represents the character and an attribute byte. This gives us the offset into the buffer. Clearing is accomplished by a block move of a blank and an attribute into the screen memory. The attribute is set in such a way that the foreground, which is the character, has all three color bits set; blinking, background, and high-intensity are turned off. The tricky part is figuring out which display card is in place, since the buffer for the monochrome monitor is in a different location from that for the color monitor. This is easily and compatibly accomplished with ROM BIOS call 11 hex, which provides the equipment configuration. Bits 4 and 5 tell us the video mode, as follows:

ETX causes the routine to return, leaving the cursor where it was. (Some of us think the cursor is sitting in front of the machine, puzzled.)

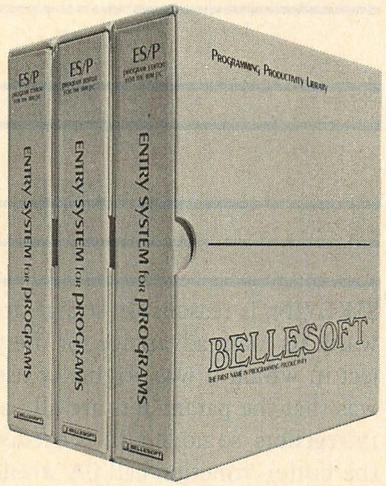
00—Unused.
01—40x25 using color card.
10—80x25 using color card.
11—80x25 using black & white card.

It would be easy to make this routine work on a forty-column screen, since the multiplier could be computed from the video mode as well.

The next function erases the current line from the cursor position to the end of the line. The method described above could be used, but this function saves the cursor position, writes blanks with function call 02, and restores the cursor position.

Handling of the bell is a bit complicated. Much of this code is taken from the IBM PC *Technical Reference* manual. The digital-to-analog channel is very poorly documented, but from what I have been able to determine, there is a timer that controls the frequency. Outputting a value to the speaker port sets that timer. Another port—the 8255 port B—deter-

BELLESOFT HAS CHANGED THE WAY PROGRAMMERS WORK:



COMPUTER AIDED PROGRAMMING

The Bellesoft Entry System for Programs (ES/P) has changed the way programmers will work in the future by putting computer aided programming (CAP) at their fingertips.

What word-processing programs did for writing text, Bellesoft has now done for generating programs: putting the computer to work to create a whole new way of writing software.

Bellesoft's ES/P speeds up software development by up to 50%, produces better programs and reduces debugging time. ES/P improves programming in five significant ways:

1. **On-Line Syntax Checking.** ES/P is more than a very smart editor: it automatically checks your program for correct syntax as you enter or change it.
2. **On-Line Formatting.** ES/P automatically provides a correct and consistent format, critical for program maintainability.
3. **Context-Sensitive Abbreviations.** ES/P builds program structures or skeletons with a single keystroke, provides correct keywords and punctuation, and saves up to half your typing time.
4. **Program Editing.** ES/P manipulates program structures correctly with a single keystroke, allows you to delete or move statements easily.
5. **On-Line Programming Help.** ES/P provides menus of correct program components and correct syntax as an aid to faster programming.

ES/P is now becoming available to computer manufacturers for inclusion in their software packages; to distributors and dealers for resale; to universities and schools as an aid to teaching programming; and to programmers who want to cut their work in half while writing better programs.*

Call your computer manufacturer or contact Bellesoft, the company that specializes in programming tools for serious programmers. Call Elizabeth Gaskill, Marketing, at (206) 828-7282, or write Bellesoft, Inc., 2820 Northup Way, Bellevue, WA 98004.

* ES/P is now available with orientation for C and Pascal. Other language orientations will be available by first quarter 1984. CIRCLE NO. 114 ON READER SERVICE CARD

BELLESOFT

THE FIRST NAME IN PROGRAMMING PRODUCTIVITY

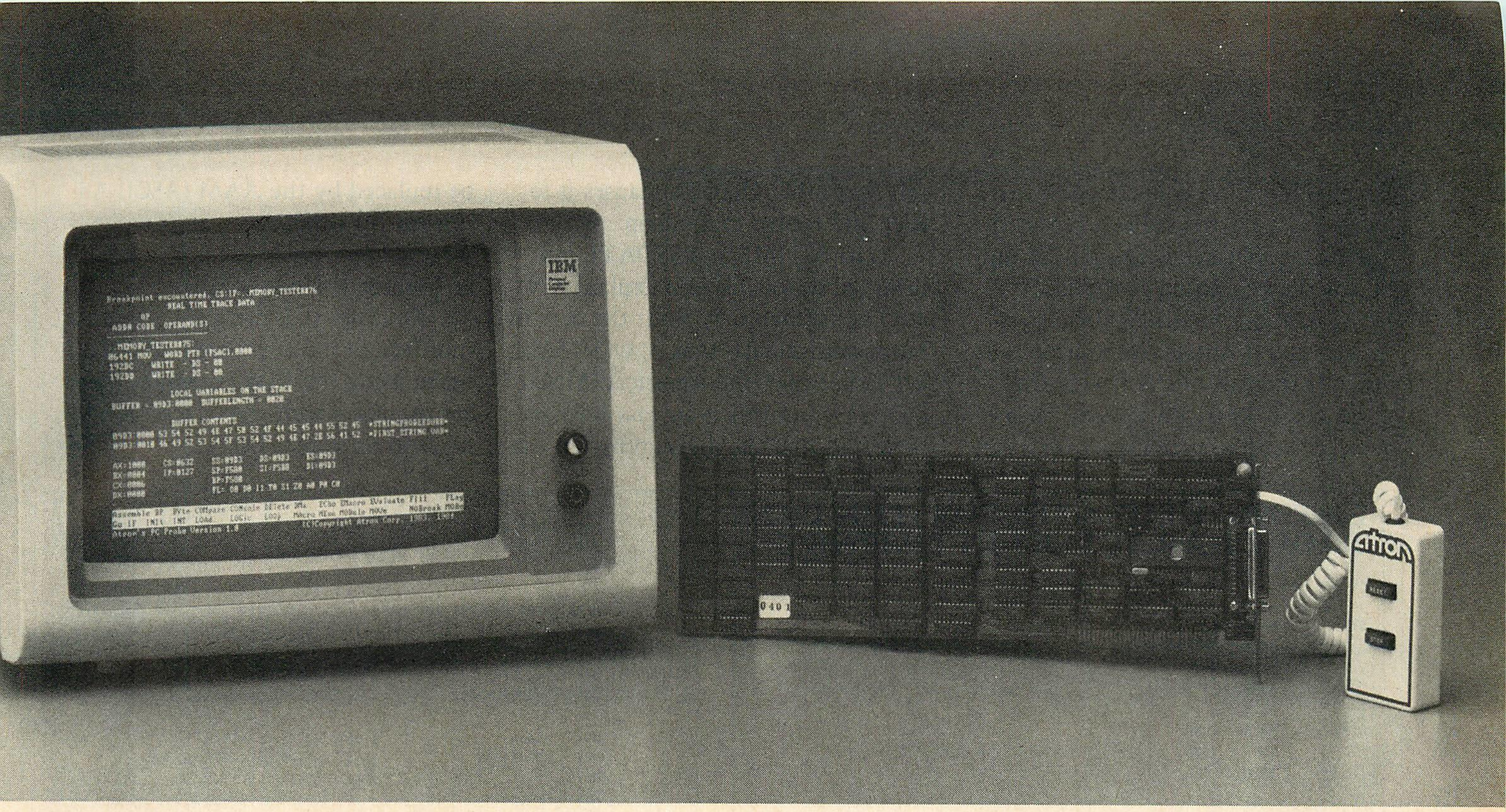
mines whether the speaker is on or off. Thus, the routine for the beep selects the frequency and then turns on the speaker. Timing for shutting the speaker off is provided by a hard loop. Note that the BASIC SOUND statement makes use of the timer interrupt to determine how long to leave the speaker on. This method does not give very fine control over the duration of the beep. In this routine the time can be controlled down to a couple of instruction cycles.

The ETX and the carriage return are handled similarly. ETX causes the routine to return, leaving the cursor where it was. (Some of us think that the cursor is sitting in front of the machine, puzzled.) The carriage-return routine writes the character to the screen after both a carriage return and a line feed have been done. If the character is not a special control, it is displayed, and the routine returns to the loop for the next one.

The keyboard-input routine, S$KEYIN, is reasonably straightforward. Its main advantage for the project in which it was originally used was that the parameters are all passed in registers, so no length is needed in the buffer; function call 0A would have required that length. This routine also allows the ESCape to leave the cursor on the same line and to act like a repeated backspace, rather than going to a new line. Note that the count passed in CX does not include the required final carriage return.

The character-input routine does not redisplay the characters in order to permit the routine to decide after the character is keyed whether it is to go to the screen. This is especially valuable for "cancel," which must not be displayed. The use of this non-displaying routine also makes the end loop much neater, since characters typed after the length has run out must never be displayed.

The only tricky part of this routine is the handling of backspace and escape. Backspace must check the current line pointer against the one that was saved on the stack, since backspacing past the beginning of the line is not allowed. The first display of a backspace is required because characters are not echoed until after they have been analyzed. The extra display of a blank is needed to get rid of the character that has been backspaced over, and the additional backspace is needed to then back up to the blank. The MS/DOS User's Guide says the backspace is destructive—that is, that the character backspaced over will be replaced by a



ATRON Announces A State of the Art Advance in Software Debugging—PC Probe

PROGRAMMERS AND MANAGERS know that finding bugs during new product development and over the entire product life cycle adds up to a significant portion of total product development cost and support time. Investing in the right debugging tools will greatly improve time to market as well as minimize development cost. Atron Corp. has the right debugging tools for the PC environment. These are:

1) PC PROBE
2) SOFTWARE PROBE
3) PERFORMANCE ANALYZER

PC PROBE plugs into a PC or compatible. It is a total system debugger with features like:

Real Time Trace
Program flow is saved in trace memory while running at full speed. PC PROBE can display trace data as high level language line numbers, procedure names etc. — or as 8088 instructions. In addition, DMA cycles, interrupt lines and external logic probes can be traced. Real time trace answers the question "How did I get here"?

Memory Protection
What good is a debugger that can be wiped out by an undebugged program? PROBE software is write protected and can't be changed.

Hardware Breakpoints
The PC PROBE has 8 breakpoints and can trap conditions such as instruction execution, read, write, IO, DMA, interrupt, or external logic probes. Breakpoints can also be set on ranges of address or data — symbolically too!

Enhanced Human Interface
The PC PROBE designers know the importance of EASE OF USE. The PC PROBE interface has a menu window which displays the syntax of each command — so you never have to remember how a command works. It also recalls the previous invocation of each command to save tedious typing — and tedious thinking!

Symbolic Debugging
Avoid the tedium of sifting through link maps to find out where things are. The PC PROBE uses your program symbols.

Macro Commands
Why be limited by a fixed set of debugging commands? PC PROBE lets you create your own powerful macro commands with parameter passing, nesting, LOOPING and IF/THEN/ELSE control.

ANNOUNCING SOFTWARE PROBE

The same great software used on PC PROBE is now available separately as SOFTWARE PROBE —

ONLY $295

Software probe is the only software debugger available which provides a hardware reset and break box for program crash recovery. What good is a software debugger if you can't get control of run away programs?

PERFORMANCE ANALYZER
How do you find time critical program problems or know where to start performance tuning your software? Get Atron's new Software Performance and Timing Analyzer. Then you can display historgrams of how your programs run — by time or by events. You can perform many different timing measurements.

Atron has many happy customers who have made critical product schedules because of PC PROBE. Why waste time on primitive debugging techniques? — Call us today and ask for your 12-page data sheet. Manuals also available for $25.

atron

20665 FOURTH STREET • SARATOGA, CA 95070 • (408) 741-5900

blank—but that seems to be untrue here. Escape is treated as a repeated backspace that keeps going back until it hits the beginning of the line.

A carriage return is always stored, and it causes the routine to return. If the full line is keyed, the routine sits in a loop waiting for a carriage return, a backspace, or a cancel. Any other character causes it to beep and return to the loop.

The S$TIME and S$DATE routines are reasonably self-explanatory. An Englishman may want the date in European format—that is, DD/MM/YY rather than the American MM/DD/YY—but this is an easy change. Note also that the division by ten can be replaced by the AAM (ASCII Adjust for Multiplication) instruction, which does exactly what is needed and which is slightly faster.

Routines that provide a standard interface to the keyboard and display make using the IBM PC easier; they are therefore well worth the time it takes to implement them.

LISTING 1 EQUATES FOR THE CONTROL CHARACTERS IN LISTING 3

```
Macros Used in Clock/Timer Subroutines

SVC         MACRO  FUNC
            MOV    AH,FUNC
            INT    21H
            ENDM
;
BIOS        MACRO  FUNC
            INT    10H
            ENDM
```

LISTING 2 MACROS USED IN CLOCK/TIMER ROUTINES

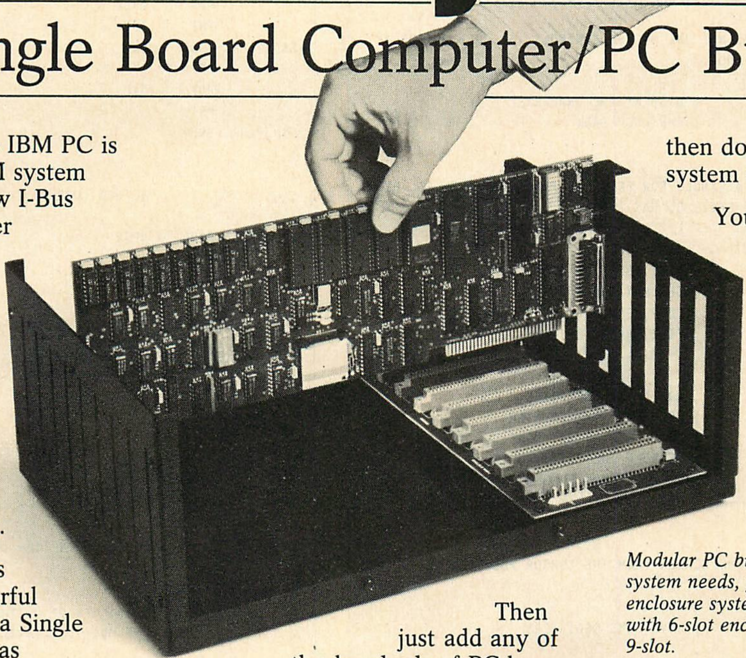
```
;Equates for the Control Characters Used in Listing 3
;
;
C$NUL     EQU    00H              ; NULL CHARACTER
C$EOS     EQU    03H              ; DISPLAY STRING TERMINATOR
C$BELL    EQU    07H             '; BELL CHARACTER
C$CPOS    EQU    09H              ; CURSOR POSITION (H,V) FOLLOWS
C$CR      EQU    0DH              ; END WITH CARRIAGE RETURN, LINE FEED
C$LF      EQU    0AH              ; LINE FEED
C$BS      EQU    08H              ; BACKSPACE
C$EEOF    EQU    011H             ; ERASE TO END OF FRAME
C$EEOL    EQU    012H             ; ERASE TO END OF LINE
C$RU      EQU    013H             ; SCROLL THE SCREEN UP
C$RD      EQU    014H             ; SCROLL THE SCREEN DOWN
C$ESC     EQU    01BH             ; ESCAPE/CANCEL
```

LISTING 3 INTERF.ASM-KEYIN/DISPLAY/CLOCK INTERFACE ROUTINES

```
;               Keyin/Display/Clock Interface Routines.
;
;       These routines provide a standard interface to the keyboard and
;       display of the IBM personal computer.
;
;       Copyright 1983 by John Cole, Duncanville, Texas.
;
;
;       This routine is the interface to the display of the machine.  It
;       interprets a string of characters terminated by either a C$EOS
;       character or a carriage return, sending the non-control characters
;       to the display, and interpreting the controls to perform various
;       functions.
;
;       ON ENTRY:
;           SI -- ADDRESS OF STRING TO BE DISPLAYED.
;
;       ON EXIT:
;           SI -- POINTS PAST LAST CHARACTER DISPLAYED.
;           DL -- INDETERMINATE.
;           AL -- INDETERMINATE.
;
S$DSPLY   PROC   NEAR              ; DISPLAY PROCEDURE
          PUSH   BX                ; SAVE REGISTERS
;
S$DSPLY0:
          MOV    DL,[SI]           ; GET THE CHARACTER
          INC    SI                ; MOVE THE POINTER
          CMP    DL,C$EOS          ; SEE IF END OF STRING
          JNE     S$DSPLY1         ; BRANCH IF NOT
          POP     BX               ; RESTORE REGISTERS
          RET                      ; ELSE EXIT
;
;         HERE CHECK FOR THE VARIOUS CONTROL'CHARACTERS, AND TAKE APPROPRIATE
;         ACTION IF WE GET THEM.  THE FIRST CODE IS SCROLL UP.
;
S$DSPLY1: CMP    DL,C$RU           ; SEE IF ROLL UP
          JNE    S$DSPLY1A         ; IF NOT, TRY SOMETHING ELSE
;
          MOV    CX,0              ; SET TOP OF SCREEN
          MOV    DX,2479H          ; SET BOTTOM OF SCREEN WINDOW
          MOV    BH,7              ; ATTRIBUTE OF BLANK LINE
          MOV    AX,601H           ; SET SCROLL CODE FOR BIOS, 1 LINE
          BIOS
          JMP    S$DSPLY0          ; LOOP BACK FOR NEXT CHARACTER
;
;         CHECK FOR SCROLL DOWN.
;
S$DSPLY1A:
          CMP    DL,C$RD           ; CHECK FOR ROLL-DOWN CODE
          JNE    S$DSPLY2          ; JUMP IF NOT
          MOV    CX,0              ; SET TOP POSITION
          MOV    DX,2479H          ; AND BOTTOM OF SCREEN
          MOV    BH,7              ; ATTRIBUTES OF BLANK LINE
          MOV    AX,701H           ; SCROLL 1 LINE, BIOS CODE IN AH
          BIOS                     ; DO IT
          JMP    S$DSPLY0          ; BACK FOR NEXT CHARACTER
;
;         CHECK FOR CURSOR POSITONING HERE.
;
S$DSPLY2:
          CMP    DL,C$CPOS        ; CHECK IT
          JNE    S$DSPLY3         ; JUMP IF NOT EQUAL
          MOV    AH,2             ; FUNCTION NUMBER
          MOV    DX,[SI]          ; GET POSITION
          ADD    SI,2             ; MOVE PAST THE POSITION
          MOV    BH,0             ; SET DISPLAY PAGE
          BIOS                    ; CALL BIOS FOR DISPLAY CONTROL
          JMP    S$DSPLY0         ; LOOP BACK FOR NEXT CHARACTER
;
;         CHECK FOR CARRIAGE RETURN, WHICH DOES A LINE FEED AS WELL. THIS IS
;         ALSO TREATED AS A STRING TERMINATOR.
;
S$DSPLY3:
          CMP    DL,C$CR          ; CHECK FOR CARRIAGE RETURN
          JNE    S$DSPLY4         ; JUMP IF NOT
          SVC    DSPCHAR$         ; ELSE PUT OUT THE CHARACTER
          POP    BX               ; RESTORE REGISTERS
          RET                     ; AND EXIT
;
;         ERASE SCREEN FROM CURRENT CURSOR POSITION TO END OF FRAME.  THE
;         LOGIC IS TO READ THE CURRENT CURSOR POSITION, THEN WRITE BLANKS TO
;         THE SCREEN FROM THERE TO THE END OF THE SCREEN.  FINALLY, WE RESET
;         THE CURSOR POSITION TO ITS FORMER VALUE.
;
S$DSPLY4:
          CMP    DL,C$EEOF        ; CHECK FOR ERASE TO END OF FRAME CODE
          JNE    S$DSPLY5         ; IF NOT, TRY NEXT ONE
;
          MOV    AH,3             ; CODE FOR READING THE CURSOR POSITION
          BIOS                    ; GET IT
          PUSH   DX               ; AND STACK IT
          CALL   S$SCLR           ; CLEAR THE REMAINDER OF THIS LINE
          POP    DX               ; RESTORE ORIGINAL CURSOR POSITION
          MOV    AH,2             ; SET ORIGINAL CURSOR POSITION
          BIOS
```

Put the Power of the IBM PC Bus into Your OEM System

NEW Single Board Computer/PC Bus System



The power of the IBM PC is now available to OEM system designers with the new I-Bus Single Board Computer and Enclosure systems. Now you can make use of that vast array of PC-compatible expansion cards—for communications, graphics, data acqui sition, peripheral control, and every other imaginable task.

I-Bus Systems has packaged Intel's powerful new 80188 CPU into a Single Board Computer. It has 64K of RAM and up to 160K of ROM on board, plus a serial console port to talk to a terminal or a PC. Just plug the SBC into an I-Bus 6-slot chassis or 9-slot card cage and you have the heart of a computer system, ready to run.

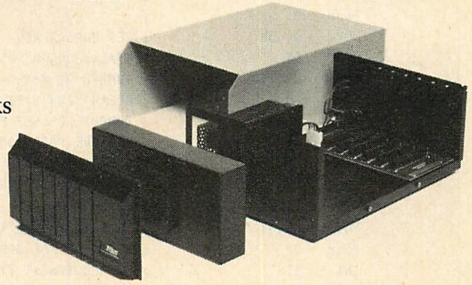
Then just add any of the hundreds of PC bus cards already on the market, to customize your system.

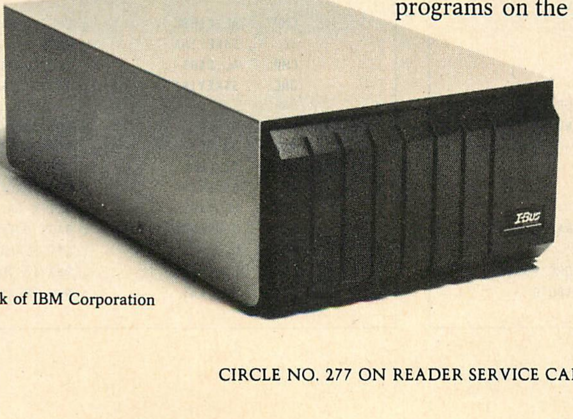
Best of all, the IBM PC works perfectly as a software develop-ment system. You can assemble and test applications programs on the PC, then download them to the I-Bus system for dedicated execution.

You'll have the best of both the PC world and the OEM bus-structured world with this new single board computer from I-Bus.

For all the details, give us a call today at (800) 382-4229. In California, call (619) 569-0646.

Modular PC bus packaging matches your system needs, from basic cage to complete enclosure system. Internal 40W power supply with 6-slot enclosure, external supply for the 9-slot.





I-BUS SYSTEMS
9235 Chesapeake Drive
San Diego, CA 92123

IBM PC is a trademark of IBM Corporation

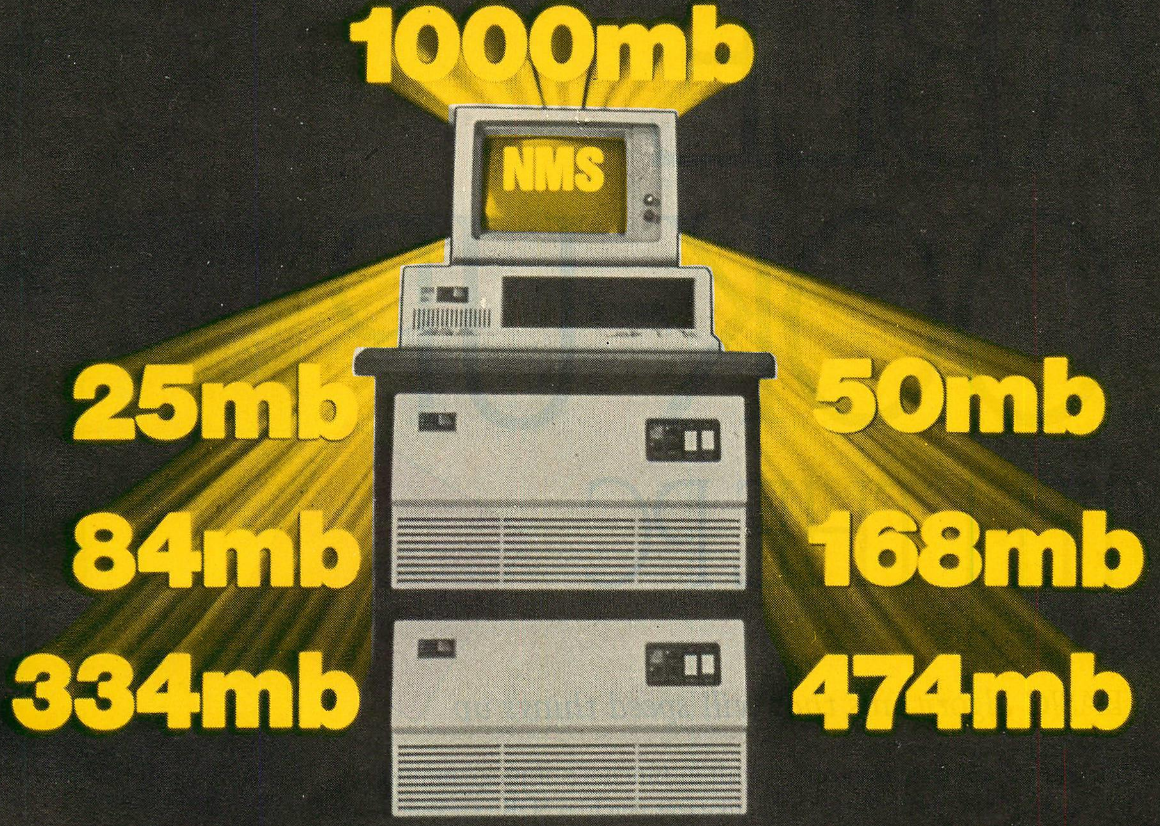
CIRCLE NO. 277 ON READER SERVICE CARD

```
        JMP     S$DSPLY0            ; AND LOOP BACK
;
;       CHECK FOR ERASE TO END OF LINE CODE.  IF SO, JUST CALL THE ROUTINE
;       TO CLEAR THE END OF THE LINE, THEN RESTORE THE CURRENT CURSOR.
;
S$DSPLY5:
        CMP     DL,C$EEOL           ; CHECK FOR THE CODE
        JNE     S$DSPLY8            ; IF NOT, TRY SOMETHING ELSE
;
        MOV     AH,3                ; READ CURSOR POSITION
        BIOS                        ; GET IT
        PUSH    DX                  ; AND STACK IT
        CALL    S$DCLR              ; CLEAR THE REST OF THIS LINE
        POP     DX                  ; RESTORE OLD CURSOR POSITION
        MOV     AH,2                ;
        BIOS                        ;
        JMP     S$DSPLY0            ; AND BACK TO THE MAIN LOOP
;
;       CHECK FOR BELL CHARACTER, CAUSES BEEP.
;
S$DSPLY8:
        CMP     DL,C$BELL           ; TRY BELL CHARACTER
        JNE     S$DSPLY9            ; JUMP IF NOT
;
        CALL    S$BEEP              ; BEEP THE SPEAKER
        JMP     S$DSPLY0            ; AND RETURN TO THE LOOP
;
;       HERE THE CHARACTER IS NOT ONE OF THE CONTROLS, SO JUST OUTPUT IT
;       TO THE SCREEN, THEN LOOP.
;
S$DSPLY9: SVC   DSPCHAR$            ; DISPLAY THE CHARACTER
        JMP     S$DSPLY0            ; AND LOOP BACK
;
;
;       SUBROUTINE TO CLEAR A SINGLE ROW FROM THE POSITION IN DL TO THE
;       END OF THE LINE.  ON EXIT, DL HAS ZERO.
;
S$DCLR:
        PUSH    DX                  ; SAVE POSITION OVER LOOP
        MOV     DH,80               ; SET MAXIMUM VALUE
        SUB     DH,DL               ; COMPUTE NUMBER OF COLUMNS TO CLEAR
        MOV     DL,' '              ; FILL CHARACTER
;
S$DCLR1:
        SVC     DSPCHAR$            ; WRITE THE CHARACTER
        DEC     DH                  ; DECREMENT THE COUNTER
        JNZ     S$DCLR1             ; LOOP BACK IF NOT DONE
        POP     DX                  ; RESTORE THE POSITION
        MOV     DL,0                ; CLEAR COLUMN NUMBER
        RET                         ; AND EXIT
S$DSPLY ENDP
;
;       SUBROUTINE TO CLEAR THE SCREEN FROM THE CURSOR POSITION TO THE END
;       OF THE SCREEN.
;
S$SCLR: PUSH    CX                  ; SAVE SOME REGISTERS
        PUSH    ES                  ;
        PUSH    DI                  ;
        MOV     AL,DH               ; GET CURSOR ROW
        PUSH    DX                  ; STACK CURSOR POSITION
        MOV     DX,160              ; NUMBER OF BYTES IN A ROW
        MOV     AH,0                ; CLEAR HIGH BYTE
        MUL     DL                  ; COMPUTE OFFSET FOR ROW
        POP     DX                  ; RESTORE POSITION
        MOV     DH,0                ; CLEAR HIGH BYTE OF COLUMN
        SAL     DX,1                ; COMPUTE IN WORDS
        ADD     AX,DX               ; COMPUTE OFFSET
        MOV     DI,AX               ; COPY TO PROPER REGISTER
        PUSH    DI                  ; STACK BUFFER OFFSET
        INT     11H                 ; GET EQUIPMENT TYPE
        MOV     DI,AX               ; COPY IT
        MOV     AX,0B800H           ; SEGMENT ADDRESS FOR COLOR CARD
        AND     DI,30H              ; SEE IF COLOR OR BLACK AND WHITE CARD
        CMP     DI,30H              ;
        JNE     S$SCLR1             ; JUMP IF COLOR
        MOV     AX,0B000H           ; SET UP BLACK AND WHITE
S$SCLR1: POP    DI                  ; RESTORE BUFFER OFFSET
        MOV     ES,AX               ; SET UP CARD BASE AS EXTRA SEGMENT
        MOV     CX,80*25*2          ; FILL LENGTH IN WORDS
        SUB     CX,DI               ; SUBTRACT OFF LENGTH BEFORE POSITION
        MOV     AX,' '+7*256        ; FILL CHARACTER AND ATTRIBUTE
        REP     STOSW               ; STORE IT ALL
        POP     DI                  ; RESTORE REGISTERS
        POP     ES                  ;
        POP     CX                  ;
        RET                         ; BACK TO CALLER
;
        PAGE
;
;               CHARACTER DISPLAY SUBROUTINE.
;
;       THIS ROUTINE PUTS THE CHARACTER ON THE SCREEN.  HOWEVER, IF IT
;       GO PAST THE END OF THE LINE, NO WRAPAROUND TAKES PLACE.
;
S$DSPCHR PROC   NEAR                ; ENTRY POINT FOR CHARACTER DISPLAY
        PUSH    BX                  ; STACK ATTRIBUTE
        PUSH    CX                  ; AND COUNTER REGISTER
        PUSH    DX                  ;
        MOV     CX,1                ; ONLY ONE CHARACTER GOING OUT
        MOV     BH,0                ; DISPLAY PAGE IN USE
        MOV     AH,9                ; DISPLAY CHARACTER/ATTRIBUTE
        INT     10H                 ; CALL THE ROM
        MOV     AH,3                ; GET CURRENT CURSOR POSITION
        INT     10H                 ;
        CMP     DL,79               ; CHECK FOR PAST END OF LINE
        JE      S$DSPCHR1           ;
        INC     DL                  ; IF NOT, MOVE TO NEXT COLUMN
        MOV     AH,2                ; NOW SET THE CURSOR
        INT     10H                 ;
S$DSPCHR1: POP  DX                  ;
        POP     CX                  ; RESTORE REGISTERS
        POP     BX                  ; . . .
        RET                         ; AND BACK TO CALLER
S$DSPCHR ENDP                       ; END OF CHARACTER DISPLAY PROCEDURE
;
;
;               RETURN CURRENT CURSOR POSITION.
;
S$CURPOS PROC   NEAR                ; GET CURSOR POSITION IN DX
        PUSH    CX                  ; STACK OVER CALL
        PUSH    BX                  ;
        MOV     BH,0                ; SET THE DISPLAY PAGE
        MOV     AH,3                ; VIDEO I/O PARAMETER
        INT     10H                 ; CALL THE ROM
        POP     BX                  ; RESTORE CALLER'S BX
        POP     CX                  ; RESTORE CALLER'S CX
        RET                         ; EXIT WITH POSITION
S$CURPOS ENDP                       ; END OF PROCEDURE
;
;
        PAGE
;
;       Routine to key in a line of a specified length into a buffer.
;
;       On entry:
;           SI -- Address of buffer.
;           CX -- Number of characters to accept.
;
;       On Exit:
;           SI -- Unchanged.
;
S$KEYIN PROC    NEAR                ; PROCEDURE ENTRY POINT
        PUSH    SI                  ; STACK BUFFER ADDRESS
;
S$KEYIN0: SVC   KEYINX$             ; GET A CHARACTER
;
;       CHECK FOR BACKSPACE AND CANCEL HERE.
;
        CMP     AL,C$ESC            ; CHECK FOR ESCAPE
        JE      S$KEYINA            ; JUMP IF SO
        CMP     AL,C$BS             ; TEST WHAT WE GOT
        JNE     S$KEYIN3            ; IF NOT, WAS SOMETHING ELSE
;
S$KEYINA: MOV   BP,SP               ; GET STACK POINTER
        MOV     DH,AL               ; COPY CHARACTER THAT GOT US HERE
S$KEYINB: CMP   SI,[BP]             ; CHECK FOR BEGINNING OF LINE
        JE      S$KEYIN0            ; LOOP BACK IF SO
        MOV     DL,C$BS             ; GET A BACKSPACE
        SVC     DSPCHAR$            ; BACK OVER CHARACTER
        MOV     DL,' '              ; GET BLANK FOR CURRENT POSITION
        SVC     DSPCHAR$            ; PUT IT OUT
        MOV     DL,C$BS             ; NOW BACKSPACE AGAIN
```

```
         SVC     DSPCHAR$        ; SINCE WE HAVE PUT OUT THE BLANK
         INC     CX              ; AND INCREMENT THE COUNTER
         DEC     SI              ; BACK UP STORAGE POINTER
         CMP     DH,C$ESC        ; CHECK FOR ESCAPE/CANCEL
         JE      S$KEYINB        ; LOOP BACK IF SO
         JMP     S$KEYIN0        ; AND LOOP BACK FOR THE NEXT ONE
;
S$KEYIN3:
         MOV     [SI],AL         ; STORE IT
         INC     SI              ; MOVE THE POINTER
         CMP     AL,C$CR         ; CHECK FOR CARRIAGE RETURN
         JNE     S$KEYIN4        ; JUMP IF NOT
         POP     SI              ; RESTORE STRING ADDRESS
         RET                     ; ELSE EXIT IF END OF LINE
;
S$KEYIN4: MOV    DL,AL           ; COPY THE CHARACTER
         SVC     DSPCHAR$        ; PUT IT ON THE SCREEN
         LOOPNZ  S$KEYIN0        ; LOOP BACK FOR NEXT CHARACTER
;
;        HERE THE COUNTER HAS DECREMENTED TO ZERO, SO WE SIT IN A HARD LOOP
;        WAITING FOR A CARRIAGE RETURN.
;
S$KEYIN5: SVC    KEYINX$         ; GET A CHARACTER
         CMP     AL,C$CR         ; CHECK FOR END
         JE      S$KEYIN6        ; EXIT IF IT IS
         CMP     AL,C$BS         ; CHECK FOR BACKSPACE
         JE      S$KEYINA        ; TAKE CARE OF THAT IF SO
         CMP     AL,C$ESC        ; CHECK FOR ESCAPE, TREATED AS CANCEL
         JE      S$KEYINA        ; AND JUMP IF SO
         CALL    S$BEEP          ; BEEP
         MOV     DL,C$BS         ; GET BACKSPACE
         SVC     DSPCHAR$        ; AND BACK OVER BAD CHARACTER
         JMP     S$KEYIN5        ; AND LOOP BACK
;
S$KEYIN6: MOV    [SI],AL         ; STORE THE TERMINATOR
         POP     SI              ; RESTORE STRING ADDRESS
         RET                     ; AND EXIT
S$KEYIN  ENDP
;
         PAGE
;        BEEP ROUTINE
;
;        THIS ROUTINE CAUSES A BEEP AT THE SPEAKER OF THE PC.
;
TIMER$   EQU     40H             ; TIMER PORT
PORT$B   EQU     61H             ; 8255 PORT B ADDRESS
;
S$BEEP   PROC    NEAR            ; START OF PROCEDURE
         MOV     AL,10110110B    ; GET TIMER SETTING
         OUT     TIMER$+3,AL     ; SET TIMER MODE
         MOV     AX,533H         ; DIVISOR FOR 1000 HZ
         OUT     TIMER$+2,AL     ; SET COUNT LSB
         MOV     AL,AH           ;
         OUT     TIMER$+2,AL     ; SET COUNT MSB
         IN      AL,PORT$B       ; GET CURRENT PORT MODE
         MOV     AH,AL           ; SAVE IT
         OR      AL,3            ; TURN SPEAKER ON
         OUT     PORT$B,AL
         MOV     CX,32762        ; COUNTER FOR ABOUT 250 MS WAIT
S$BEEP1: LOOP    S$BEEP1         ; DELAY BEFORE TURNING OFF
         MOV     AL,AH           ; RECOVER OLD PORT VALUE
         OUT     PORT$B,AL       ; SET IT THE WAY IT WAS
         RET                     ; AND EXIT TO CALLER
S$BEEP   ENDP                    ; END OF ROUTINE
         PAGE
;
;                GET TIME AS A STRING.
;
;        This routine returns a string containing the current time as read
;        from the system clock.  The string is of the form HH:MM:SS.
;
;        On entry:
;            SI -- Address into which string is to be placed.
;
;        On Exit:
;            AX -- Changed.
;
S$TIME   PROC    NEAR            ; START OF TIME PROCEDURE
         PUSH    SI              ; STACK INCOMING REGISTERS
         PUSH    DX              ; . . .
         PUSH    CX              ; . . .
         SVC     TIME$           ; GET THE TIME
```

```
         PUSH    DX              ; STACK SECONDS
         MOV     AL,CH           ; THE HOURS ARE GOOD
         CBW                     ; CONVERT TO WORD
         MOV     DL,10           ; SEPARATE THE DIGITS
         DIV     DL              ;
         ADD     AX,'00'         ; CONVERT TO ASCII
         MOV     [SI],AX         ; SAVE HOURS
         MOV     BYTE PTR 2[SI],':' ; STORE SEPARATOR
         ADD     SI,3            ; MOVE STORE POINTER
         MOV     AL,CL           ; NOW GET THE MINUTES
         CBW                     ; CONVERT
         MOV     DL,10           ; GET DIVISOR
         DIV     DL              ; SPLIT THE DIGITS
         ADD     AX,'00'         ; CONVERT TO ASCII
         MOV     [SI],AX         ; STORE IT, LSB,MSB
         MOV     BYTE PTR 2[SI],':' ; AND KEEP THEM APART
         ADD     SI,3            ; MOVE MICKEY'S BIG HAND
         POP     DX              ; RESTORE SECONDS
         MOV     AL,DH           ; SECONDS, PLEASE
         CBW                     ; CLEAR HIGH BYTE OF DIVIDEND
         MOV     DL,10           ; DIVISOR
         DIV     DL              ; SPLIT
         ADD     AX,'00'         ; CONVERT TO ASCII
         MOV     [SI],AX         ; STORE
         POP     CX              ; RESTORE CALLER'S REGISTERS
         POP     DX              ; . . .
         POP     SI              ; . . .
         RET                     ; AND GO HOME
S$TIME   ENDP                    ; END OF TIME CONVERSION
;
         PAGE
;
;                DATE CONVERSION ROUTINE.
;
;        This routine gets the system date and converts it into a string of
;        the form MM/DD/YY.
;
;        On Entry:
;            SI -- Address into which date is to be returned.
;
;        On Exit:
;            AX -- Changed.
;            All other registers preserved.
;
S$DATE   PROC    NEAR            ; DATE CONVERSION PROCEDURE
         PUSH    SI              ; STACK CALLER'S REGISTERS
         PUSH    CX              ; . . .
         PUSH    DX              ; . . .
         SVC     DATE$           ; CALL UP THE DATE
         PUSH    DX              ; SAVE THE DAY
         MOV     AL,DH           ; GET THE MONTH
         CBW                     ; EXPAND IT OUT
         MOV     DL,10           ; GET DIVISOR
         DIV     DL              ; CONVERT MONTH
         ADD     AX,'00'         ; CONVERT TO ASCII
         MOV     [SI],AX         ; STORE IT
         MOV     BYTE PTR 2[SI],'/' ; STORE SEPARATOR
         ADD     SI,3            ; MOVE TO DAY
         POP     DX              ; RESTORE FROM STACK
         MOV     AL,DL           ; WHAT DAY IS IT?
         CBW                     ; EXPAND IT
         MOV     DL,10           ; DIVISOR
         DIV     DL              ; SPLIT THE DIGITS APART
         ADD     AX,'00'         ; CONVERT TO ASCII
         MOV     [SI],AX         ; STORE IT
         MOV     BYTE PTR 2[SI],'/' ; STORE SEPARATOR
         ADD     SI,3            ; MOVE TO YEAR
         SUB     CX,1900         ; SUBTRACT BIAS FROM YEAR
         CMP     CX,100          ; SEE IF TURN OF MILLENIUM PAST
         JL      S$DATE1         ; JUMP IF NOT (HIGH PROBABILITY)
         SUB     CX,100          ; ELSE MAKE SURE WE GET JUST 2 DIGITS
S$DATE1: MOV     AL,CL           ; GET THE YEAR
         CBW                     ; CLEAR HIGH BYTE
         MOV     DL,10           ; DIVISOR
         DIV     DL              ; SPLIT
         ADD     AX,'00'         ; CONVERT
         MOV     [SI],AX         ; STORE
         POP     DX              ; RESTORE CALLER'S REGISTERS
         POP     CX              ; . . .
         POP     SI              ; . . .
         RET                     ; EXIT TO CALLER
S$DATE   ENDP                    ; THE DATE IS OVER
```

National Memory Systems' PC-8000 Series Has No Competition.



When you look for high capacity mass memory storage for your IBM PC or compatible, you will find only one company that offers reliability, proven technology, up to one gigabyte capacities and immediate availability–National Memory Systems Corporation.

LARGE DATA BASE AND NETWORKING APPLICATIONS.
Now you can have the capacity of 3000 floppies without the headaches. For large data base and accounting applications or as a file server on a local area network, the PC-8000 is ideal. In a 50-station network, a single file server could supply almost 20 megabytes of dedicated storage to each PC at a very low cost. Eliminate performance bottlenecks with the PC-8000's lightning fast access times... four times faster than an XT.

SOPHISTICATED PERFORMANCE.
The PC-8000 utilizes a single board controller. Unsurpassed performance of 17 millisecond average access times and data transfer rates of nearly 2 megabytes per second can now be yours. The PC-8000 Series supports overlapped seeks, 32-bit ECC with 11-bit error correction, high speed DMA and more. One single slot controller supports up to two disk drives with capacities ranging from 25 megabytes all the way up to 1,000 megabytes. Add our 9-track, ANSI-compatible tape backup and you've got the best performing subsystem being delivered today.

HIGH RELIABILITY. Failure is unlikely with MTBF figures in excess of 30,000 hours.

Backed with our standard 12-month warranty and nationwide maintenance service your investment is protected.

CALL US. If you're looking to buy or lease high capacity disk storage for your IBM PC or compatible, and you want delivery NOW, call National Memory Systems at (415) 443-1669. We also sell and lease high performance subsystems for DEC, DG and TI minicomputers.

National Memory Systems CORPORATION

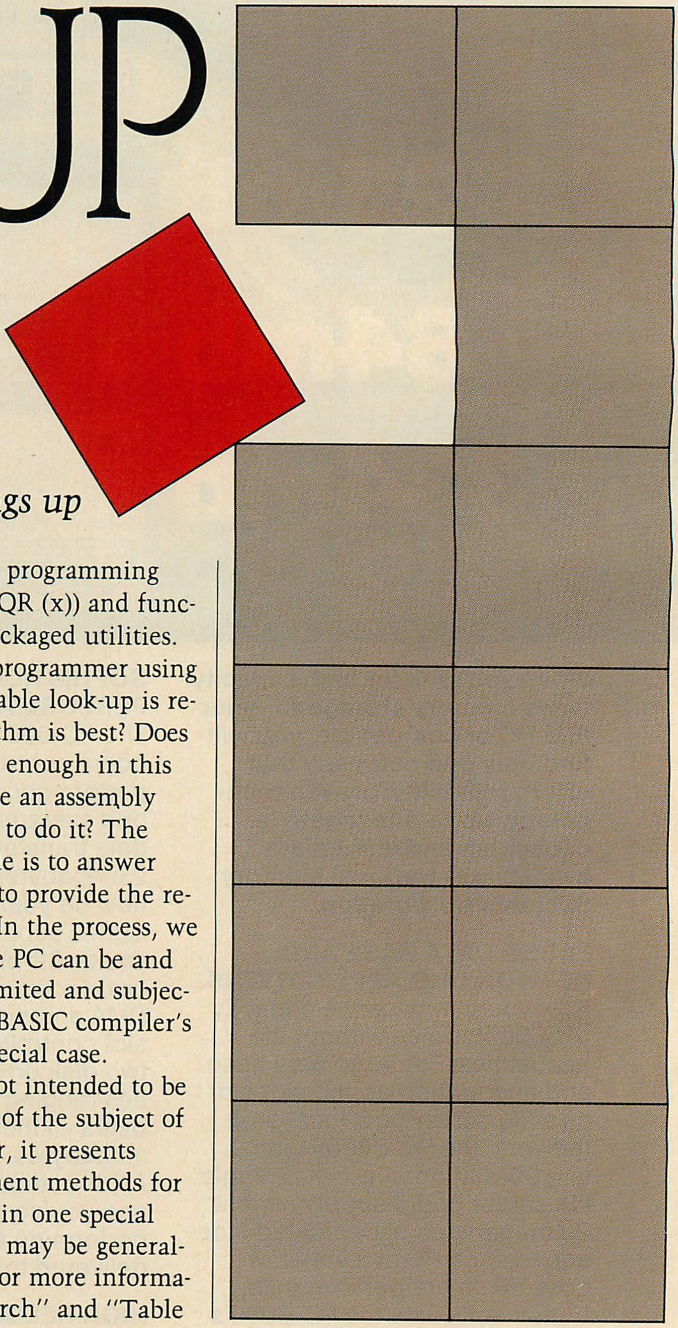
355 Earhart Way,
Livermore, CA 94550

25mb to 1,000mb capacities *NOW!*

CIRCLE NO. 111 ON READER SERVICE CARD

RALPH G. BRICKNER

TABLE LOOK-UP WITH THE PC



Three BASIC algorithms that will speed things up

As everyone knows, computers excel at repetitive tasks that require little intelligence, such as printing out invoices, sorting insurance policies, or calculating sales tax on 1,000,000 transactions. One task that is commonly required of computers is the table look-up, a completely mindless function that crops up in many applications. For example, a computer may read employee records from disk and may have to look up the employee number in a table to find the pay rate, tax information, etc. The possibilities are nearly endless, which is why table look-up is a fairly well-studied subject.

How does the IBM Personal Computer do at table look-up? A brief perusal of the programming language manuals indicates there are no built-in functions for this task, which is not really surprising. Table look-up is one of those gray areas between the built-in functions of programming languages (such as SQR (x)) and functions provided by packaged utilities.

Imagine that a programmer using BASIC finds that a table look-up is required. What algorithm is best? Does BASIC perform well enough in this area? Should he write an assembly language subroutine to do it? The purpose of this article is to answer these questions and to provide the required subroutines. In the process, we will see how fast the PC can be and get an admittedly limited and subjective estimate of the BASIC compiler's efficiency for this special case.

This article is not intended to be an exhaustive study of the subject of table look-up. Rather, it presents some easy-to-implement methods for getting the job done in one special case. These methods may be generalized by the reader. For more information, see "Binary Search" and "Table

Lookup," in the *Encyclopedia of Computer Science and Engineering*, edited by Ralston and Reilly and published by Van Nostrand Reinhold and Company. Also, the following IBM Personal Computer manuals may be helpful: BASIC, BASIC Compiler, Macro Assembler, and Disk Operating System.

Definitions

A few terms should be defined here:

Table—an ordered (i.e., sequential) list of integers occupying consecutive memory locations. Although the techniques discussed in this article are readily extended to tables of floating-point numbers or tables with non-unit "stride" (non-consecutive memory locations), the ordered character of this table is crucial to the second two of the three algorithms that follow. Since these algorithms are so much faster than the simple sequential search presented first, most applications will probably show a considerable gain in performance even if the table must be sorted first. Another restriction is that the table must reside in memory, e.g., as a BASIC array of numbers.

Key—an integer. The table is to be searched until one of the table entries is found to match the key, which unlocks the information coded in the table. The purpose of the search is to find the *position* in the table occupied by the integer that matches the key. For example, if the table consists of the integers 2, 4, 6, and 8, and the key is 6, the purpose of the search is to return the answer 3, because 6 is located in position 3. For simplicity, we will assume that the table always contains the key. That is, we will not test for the possibility that the table does not contain the key.

Look-up or *search*—the process of finding the position of the key in the table.

Index—the position a particular integer occupies in the table. For the table consisting of 2, 4, 6, and 8, the index of 8 is 4.

Compare—the act of comparing the key to a table entry. In BASIC, this might be encoded IF (ITABLE (I) > JKEY), and in assembly language it might be encoded CMP AX, [BP] [SI]. Note that the number of statements per compare differs in the algorithms presented below. That is, for the sequential search, there are fewer BASIC statements per compare than for the binary search, etc.

THE ALGORITHMS

The first algorithm for searching the table, a sequential search, is probably the first method most people would think of if asked to search a table for an integer. The entries in the table are searched one by one, beginning with the first, until a match is found. If the key is equally likely to be any of the table entries, the maximum number of compares is equal to the table size and the average number of compares is equal to one-half the table size. Although this method certainly does work, it can be very slow when used with a large table.

The second algorithm is a modified sequential search. Instead of searching by comparing each key until a match is found, the search looks at every ten entries until the entry is greater than or equal to the key. It then backtracks through the ten entries it skipped until it finds the matching item. For a table size of NTABLE, the maximum number of compares is about NTABLE/10 + 10, while the average number of compares is (NTABLE/2)/10 + 10/2. This represents nearly an order of magnitude improvement over the straight sequential search. Nevertheless, it is still an "order NTABLE" algorithm, meaning that if NTABLE doubles, so does the average number of compares.

The third algorithm is known as a "binary search," because it involves throwing away half of the table each time a compare is made. It is basically the second algorithm developed to its maximum potential. Instead of using an additive constant to find the next table entry to compare, this method uses a multiplicative constant, which is 1/2.

First, the middle of the table is

The sequential search algorithm is probably the first method most people would think of if asked to search a table for an integer.

found, and then the key is compared to the value of the middle entry. If the key is equal to this entry, the search is finished; if the key is larger than this entry, the next search will consider only the top half of the table; if the key is smaller, the next search will consider only the bottom half. If the key has not yet been matched, the middle of the remaining half of the table is found, and the process begins again, continuing until the correct entry is found.

In programming this algorithm, I used two bounds on the table elements. The first starts as 1, the first table element. The second starts as NTABLE, which is the number of elements in the table. The binary search then consists of examining the entry halfway between them, and, if that entry does not match the key, resetting the lower or upper bound.

Theoretically, the maximum number of compares needed to find a match for the binary search is about $\log_2(\text{NTABLE}+1)$ for NTABLE table entries, and the average number of compares is about $\log_2(\text{NTABLE})-1$. The binary search is thus an "order log NTABLE" algorithm, meaning that doubling the number of table entries does not double the work.

THE MAIN PROGRAM

Listing 1 is a simulation of a possible application program. The basic idea is

NEWS FOR BUSINESS USERS OF IBM MINI'S...

Announcing DBA 34/36... the RPGII Data Base

5 Reasons Why DBA 34/36 Will Increase the Facility, Speed, Accuracy and Flexibility of Your Information Processing and Lower Your New Business Applications Costs

Now there is a true data base management system (DBMS) for IBM System 34 and 36 users which organizes computer files into a single, coordinated information bank that is accessible in a variety of logical groupings—DBA 34/36 from Fits Systems Inc. Here are five reasons why you should be using it with your mini:

1. DBA 34/36 is the only data base management system designed for both the IBM System 34 and the new System 36—a true data base, not just a scaled-down version of mainframe software or retrieval tool.

2. DBA 34/36 makes it easy to define and produce recurring reports including simple user-defined calculations. All requisite support programs, including Data Base Maintenance, a powerful Universal Inquiry Program. RPG II Interface and a Report Generator come with the package.

3. Because DBA 34/36 is written in RPG II, your programmers can adapt any of its calling subroutines into your own programs, significantly reducing programming time for new applications.

4. DBA 34/36 can be up-and-running in less than a week because it does not change existing files. Instead, DBA 34/36 creates an external structure that manages references to your files with remarkable efficiency. An experienced programmer can install DBA 34/36 and have it running inquiry against your files in just a few days.

5. No schooling is required—your staff will quickly become proficient with DBA 34/36 by simply using the "user friendly" manual supplied with the package. The manual describes installation procedures and includes complete instructions for tailoring DBA 34/36's functions to reference the files selected for installation.

You Can Try DBA 34/36 For 45 Days FREE

We offer qualified users a free 45 day trial of DBA 34/36 in order to demonstrate what it can do for you. We'll send you a copy of the software, and complete installation documentation, for a 45 day free trial to use on your own system. We think you'll wonder how you ever did without it—but, if you're not completely satisfied, simply return the package—there's no further obligation.

Free Demonstration Disk Available

FITS

Casey Stern, Vice President
FITS Systems, Inc.
DBA Department
65 West Red Oak Lane
White Plains, N.Y. 10604
212-269-5540 • 914-694-1554

☐ YES, I'm interested in your DBA 34/36 Software Package.

Please send me:
☐ FREE DEMO DISC which illustrates DBA 34/36
☐ COMPLETE PACKAGE for 45-day free trial
☐ MORE INFORMATION—have a sales rep contact me.

Name__________ Title__________
Company__________
Address__________
City__________ State__________ Zip__________
Telephone ( )__________ Best Time__________

Mail This Coupon Or Call (914) 694-1554

CIRCLE NO. 249 ON READER SERVICE CARD

to generate random numbers that fall within the range of a table of integers and to look up those random numbers in the table. The table in this case consists of the integers themselves; the first entry is "1," the second "2," and so on. Of course, there is no real need to look up the key in the table, since the key is its own index, but this table is easy to generate and serves the purpose well enough.

The program asks the user for the table size and the number of look-ups per "run." A "run" is a set

To do the absolute timing runs, a subroutine (GOSUB 7000) reads the system time by means of the TIME$ variable and converts the minute and second parts of the string to a floating-point number.

of look-ups; there are five runs per program execution, each with a different random-number seed. The purpose of this is not only to give an average number of compares or seconds per look-up but also to give the user some idea of the variation in these numbers from run to (same-size) run.

Thus, entering "4000" and "500" will generate five sets of output, one for each of the random-number seeds, and a final overall average of the five runs. The sets of output will correspond to five runs of 500 integers, chosen at random, each looked up in a table of 4000 integers. I have used the main program in two versions. The first includes counters to measure the number of table compares in each run, but in the second version, these counter statements are REM'ed out for the timing runs. This is because floating-point arithmetic operations take a considerable amount of processing time in BASIC; because they are not intrinsically a part of the table look-up process, they would skew the timing results. Listing 1 is this second version. To return the counters, just find all strings "REM CMP" and change them to " ".

To do the absolute timing runs, a subroutine (GOSUB 7000) reads the system time by means of the TIME$ variable and converts the minute and second parts of the string to a floating-point number. After the look-ups have been done, the program calls the subroutine at 7000 again and takes the difference of the two numbers found. The result is the number of seconds it took to do the task.

The BASIC program in listing 1 generates the table (ITABLE), and picks random numbers (JKEY) to find in the table. Note that it is written for the IBM PC BASIC compiler, using the /N option. This allows line numbers to be omitted except where necessary, for greater readability. To run this on the BASIC interpreter, line numbers must be inserted in sequence. Note also that the program as written uses the $INCLUDE compiler metacommand to insert the specific subroutine to be tested. This is to save diskette space and to allow changing only one file when updating the main program. The INCLUDED subroutine is called by the statement GOSUB 8000; it does one look-up per call. For the interpreter, the MERGE command could be used to accomplish what the $INCLUDE metacommand does here.

Since an assembly language subroutine is also used. The interpreter user should also note that the procedure for linking assembly language to compiled BASIC is considerably different than that for linking machine language to the interpreter. Refer to Appendix C of the BASIC manual for a complete discussion of this subject.

Henceforth, I will assume that the user has the BASIC compiler, the macro assembler, and, as part of the operating system, the linker. However, listing 4 is a DEBUG unassembly of the assembler subroutine, and anyone using the interpreter may refer to this listing in order to POKE the machine code (column two of the listing, in hexadecimal) for use in an interpreter program. In this subroutine there are no absolute addresses that need to be converted.

THE SUBROUTINES

Listing 2 is a set of four subroutines: sequential search, in BASIC; modified sequential search, in BASIC; binary search, in BASIC; and BASIC call to assembly language binary search. Listing 3 is the assembly language coding for the binary search called from BASIC. Note that this subroutine makes no attempt to optimize the code, being instead a literal translation of the BASIC coding. There are enough 8088 registers that in the main loop the program needs to refer to memory only for the value of the table entry. This saves time, since instructions with memory references generally take longer to execute than those with register references. Listing 4 is the unassembly of this subroutine for interpreter users.

To compile and link these programs and subroutines, assume that the main program is named MAIN.BAS on drive A:; the assembler subroutine is named BINS.ASM on drive A:; and the assembler, compiler, BASRUN.LIB, and linker are located on drive B:. Here is a batch file that does all the necessary work:

```
B:BASCOM MAIN/N,
  B:MAIN.OBJ, NUL
B:MASM BINS, B:BINS.OBJ, NUL
B:LINK B:MAIN+B:BINS, MAIN,
  CON, B:BASRUN
```

If this file is called CAL.BAT (Compile, Assemble, Link), simply typing "CAL" will run the batch file. The result will be a file called MAIN.EXE on drive A:. To run it, just type "MAIN". (Since the compile was made without the "/0" option, the file BASRUN.EXE must be used to run the program.)

By specifying CON for the linker listing, the user can see where the assembler subroutine is loaded, under

the Start column of the listing that appears on the screen. The assembly language subroutine is loaded in segment ASM-SEG, as specified in the SEGMENT statement in listing 3.

RESULTS

Table 1 gives the average number of compares required by each of the three methods to find the key in the table. I used 500 as the number of look-ups per run, and the table shows the results for table sizes of 8, 80, 800, and 8000. As expected, the sequential search takes about NTABLE/2 compares per look-up; the modified sequential search takes about ten times fewer. The binary search not only is incredibly efficient compared to the first two methods but takes only three additional compares each time the table size is multiplied by ten. Such is the way of "order log NTABLE" algorithms.

Table 1 does not give us the bottom line, however, since an examination of the subroutines in listing 3 shows that the sequential search has considerably less coding per compare than the other two. Could it be that it really takes less time, even though it does more work? Table 2 shows that this is not true.

For small tables, just about any look-up procedure will do, and the simplicity of the sequential look-up has much to recommend it. For large tables and large numbers of look-ups, the binary search is clearly the way to go.

The results in this table were obtained by running 10,000 look-ups per run instead of 500. The Dummy row is the time the program took with the look-up subroutine replaced by a simple RETURN statement, and the Binary, Assembler row gives the results for the assembler subroutine. The row with the Dummy subroutine reveals the overhead involved in running the program. The other four rows give running times from which the Dummy time has been subtracted. These are thus the actual times spent in the subroutines.

There is clearly no question which method is best: the binary search wins hands down for large tables. For small tables, just about any of the algorithms will do, unless many look-ups are to be done.

It might seem surprising how fast even the slowest method is, in absolute terms. After all, it would take a human being quite some time to look up 10,000 keys in a table of 800, but even the sequential search does this in only three and a half minutes. But the 8088 in the PC runs at 4.77 MHz, and, with the average machine instruction taking perhaps 15 clock cycles to execute, about 300,000 instructions per second can be executed. As the assembler subroutine reveals, there aren't all that many instructions involved in doing the look-up; the computer can therefore execute these routines incredibly quickly.

It might also seem surprising that the assembler subroutine is only about four times faster than the BASIC binary search. Remember that the assembler subroutine was not optimized, being instead a literal translation of the BASIC subroutine. However, even with the utmost optimization, this subroutine would probably run no more than 20 percent faster.

As another data point to consider, at the other end of the computing spectrum, the CFT FORTRAN compiler running on the CRAY-1 supercomputer will, as a rule, generate code about three times slower than hand-written, highly optimized assembly language code. (This is true only for what is known as "scalar" code.) The BASIC compiler isn't doing too badly by comparison.

More germane to the discussion is how this ratio compares with other compilers running on micros; I invite ambitious readers to investigate.

RATINGS

For small tables, just about any look-up procedure will do, and the simplicity of the sequential look-up has much to recommend it.

For large tables and large numbers of look-ups, the binary search is clearly the way to go. The assembly language incarnation of the binary search, however, is probably appropriate only for those applications demanding the utmost in performance, an area in which few PCs are likely to operate. If this extra margin of performance is needed, however, it's nice to know that the IBM PC is capable of providing it.

Ralph Brickner is a physicist who develops scientific programs at the Los Alamos National Laboratory in New Mexico.

Table 1: Average Number of Table Comparisons Required for Match

	Number of Table Entries			
Method	8	80	800	8000
Sequential	4	40	394	3935
Mod. Sequential	4	11	46	400
Binary	3	6	9	12

Table 2: Time Required for 10,000 Matches (in seconds)

	Number of Table Entries			
Method	8	80	800	8000
Dummy	19.60	19.60	20.20	20.00
Sequential	2.60	21.40	211.80	2103.00
Mod. Sequential	3.40	6.80	30.20	271.00
Binary	3.40	7.80	12.20	17.00
Binary, Assembler	1.80	2.40	3.40	4.40

PC NEWS FROM CMC

Personal Telephone Management

Mainframe Technology at the PC Level

Written by Isaac M. Alderidge

BELLEVUE, WA—A manager plans to use it to track progress of his sales people while he's away on business trips.

A research center will use it to do telephone polling.

A stockbroker says she can use it to call her clients with the day's market trends and take orders after hours.

No, it isn't an executive secretary. And it isn't a hard-working telephone receptionist.

It's a system known as personal telephone management and several industry analysts are claiming this type of application could become one of the main uses of the personal computer in the near future.

Enthusiastic Response

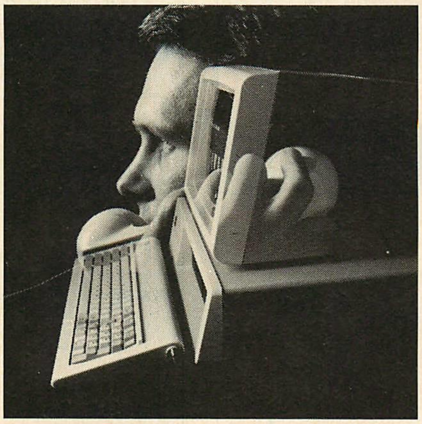
CMC International, a five-year-old public company specializing in peripheral products for the IBM® PC has introduced what it believes to be the first such system for the personal computer. After two years in development and testing the company is set to ship its first pc-DIAL/LOG™ production models. The product received an enthusiastic welcome at its recent unveiling to analysts and the press.

"Almost everyone who has seen the product comments on how they can use pc-DIAL/LOG," says Gerald Jones, CMC International executive vice president. "Acceptance is unbelievable. Computer stores are anxious to get pc-DIAL/LOG on their shelves. Many OEM and system houses have signed orders and are rushing to bundle the product with pc-based hardware. People realize the impact this type of technology brings to the pc market."

CMC's Entry

The CMC device, an expansion-type circuit board with software is designed to run on the IBM® PC or compatible computers. With pc-DIAL/LOG installed the computer is able to perform a number of telephone functions including digitally recording, saving, and sending voice messages. CMC's method allows exact reproduction of the user's voice.

Priced at under $1,000, the product offers capabilities similar to much and more complicated systems which range from $2,500 to $50,000. Although primarily designed for business, its low cost also makes it practical for home use.



Much of CMC advertising centers around this visual depicting the fusion of telephone, voice and the IBM PC.

Huge Consumer Base

Markets include the medical, legal and accounting professions, stock brokers, sales organizations, and a variety of small companies and Fortune 1000 departments, according to market studies.

Applications include message distribution, appointment reminders, telephone marketing, polling, and research.

"We see this as a tool to increase the productivity of anyone who uses the telephone a lot," says Jones.

Efficiency experts have long recognized that "phone tag" is the biggest single waste of an executive's time. Research shows only about a third of business calls reach the intended party on the first try.

"Personal telephone management has exciting potential for early and widespread acceptance very much like word processing for micros had five years ago," says Jones. "It's a natural and logical step in the evolution of personal computer usage."

Powerful

Take a task as mundane as that of notifying club members of an upcoming meeting as an example of pc-DIAL/LOG's electronic wizardry.

Press a few buttons and the peripheral calls the list of participants with a message informing them of the date. It makes its calls within certain predetermined hours and before a certain deadline. It keeps trying to reach people until successful or until instructed to stop. It also logs and organizes responses by recognizing the touch tones each person inputs —for instance, "1" if they can make it and a "3" if they'll be absent.

It will not only answer the phone, but can let specific callers with special ID numbers access individual messages. It can screen calls letting only certain persons through. Plus, it can even forward important calls to a remote location.

Requirements

CMC's pc-DIAL/LOG requires an IBM® PC or equivalent with 192K bytes of RAM and dual floppy disks. A telephone is used for recording. The package comes complete with hardware, software, manual, necessary phone cables and a telephone key-pad guide for remote access of functions.

Hard disk subsystems like CMC's TARGA™ or hard disk-equipped computers like the IBM® XT can further enhance the peripheral by providing more space for storing voice messages.

CMC's device is designed for simple plug-in installation. Software is easy to understand. Operation is menu driven which means all functions are displayed on the screen at the same time.

At a glance, a pc-DIAL/LOG owner sees who has called, at what time, what messages were sent and received, how much recording time remains and so forth.

Continuing R & D

Company officials indicate the pc-DIAL/LOG marks CMC's further diversification into advanced pc technology. The company will announce more peripherals in coming months but declined to be more specific.

Meanwhile, CMC believes they have a winner and the enthusiasm of another electronics company executive seems to validate their belief.

"It's neat! pc-DIAL/LOG frees me from the telephone when I have more important things at hand. It makes me more productive."

For information on pc-DIAL/LOG contact CMC International at 1-800-262-4685. (1-206-885-1600 in Washington.) Reprints of this article are available from CMC. ■

LISTING 1 TABLE LOOK-UP TEST PROGRAM

```
'
'    Table look-up test program.
'    For IBM Personal Computer BASIC Compiler
'    (/N Compiler option). Ralph G. Brickner  9/83
'
'    The purpose of this program is to test
'    various table look-up algorithms.  We first
'    set up a simple table of NTABLE entries.  To
'    simulate an application program, we generate
'    random numbers and find their place in the
'    table by means of a GOSUB statement.  The
'    target subroutine is changed to test the
'    various algorithms.  There is an outer loop
'    that resets the random number seed.  After
'    each "program" of NLOOK look-ups, timing or
'    compare statistics are printed out.  After
'    five such "programs", each with a new random
'    number seed, final timing or compare statistics
'    are printed.
'
'!!!!!!!!!!!!!
'Problem set-up
'!!!!!!!!!!!!!
'
     defint i-n           'Follow FORTRAN convention.
     dim itable(8000)     'Our test table, maximum of
'                         '8000 entries.
'
'    Get number of table entries we will use:
'
     input "Enter number of table entries:  ", ntable
'
'    Get number of look-ups per run:
'
     input "Enter number of look-ups per run:  ", nlook
'
     lprint chr$(10) + chr$(10) + "Table look-up
        benchmark results."
     lprint using "##### Table Entries, ##### Look-ups
        per run";ntable;nlook
'
'    Our test table consists of NTABLE integers,
'    with ITABLE(i) = i.
'
     for i=1 to ntable
     itable(i) = i
     next i
'
     timetot = 0          'Total run time tally.
'
REM CMP   cmptot = 0      'Total compares tally.
'
     time$ = "00:00:00"   'Set problem start time
'
'1111111111111
'Outer loop
'1111111111111
'
     for iseed=1000 to 5000 step 1000
'
       randomize (iseed) 'Set random number seed.
'
       gosub 7000         'This subroutine gets the
                          'system time and converts
                          'to single precision.
       time1 = t.         'Save start time for this run.
'
REM CMP    cmpr = 0       'Number of compares in this run.
'
'2222222222222
'Inner loop
'2222222222222
'
       for iloop = 1 to nlook
'
          jkey = int (rnd*(ntable)) + 1  'Number whose
                                         'index we want.
'
          gosub 8000
'
REM CMP      cmpr = cmpr + cmp           'CMP = number of
                                         'compares this look-up.
'
        next iloop
'
'2222222222222
'End inner loop
'2222222222222
'
'     Clean up for this random number seed.
'
        gosub 7000                 'This subroutine gets the
                                   'system time and converts
                                   'to single precision.
        time2 = t.
        time  = time2 - time1      'Calculate time for this run.
'
        lprint using                    "Seed: ####"; iseed;
        lprint using                 "  Time: ######.##"; time
REM CMP lprint using    "   Avg. Compares This Run: #####"; cmpr/nlook
'
        timetot = timetot + time
'
REM CMP     cmptot = cmptot + cmpr
'
      next iseed
'
'1111111111111
'End outer loop
'1111111111111
'
'     Final clean-up:
'
      lprint using "Total time used:#######.##"; timetot
      lprint using "Average time per run: ######.##"; timetot / 5
REM CMP lprint using "Average Compares: #####"; cmptot / (5. * nlook)
'
      end
'
'$$$$$$$$$$$$$
'Subroutine
'$$$$$$$$$$$$$
'
'     This subroutine calls the function TIME$ to get
'     system time.  It then converts the minute and second
'     entries to a single-precision number.
'     This value is returned in the variable "t."
'
7000 temp$ = time$
      t. =              val( mid$( temp$, 8, 1) )
      t. = t. + 10.  * val( mid$( temp$, 7, 1) )
      t. = t. + 60.  * val( mid$( temp$, 5, 1) )
      t. = t. + 600. * val( mid$( temp$, 4, 1) )
'
      return
'
'>>>>> Insert the particular search subroutine here:
'
REM $INCLUDE: 'SEARCH.BAS'
'
```

LISTING 2 BASIC TABLE LOOK-UP SUBROUTINES

```
'
'$$$$$$$$$$$$$
'Subroutine
'$$$$$$$$$$$$$
'
'     This is the sequential search subroutine.
'     Input is the table in which to look up (ITABLE), the
'     number of entries in the table (NTABLE), and the
'     number whose index is to be found (JKEY).  Output
'     is the index (INDX) such that
'     TABLE(INDX) = JKEY
'
```

Run More Than One Program With

MULTI-JOB™

FOR THE IBM PERSONAL COMPUTER AND XT

- No special hardware is required (except memory).
- Multiple programs can run concurrently using PC DOS.
- Programs can be run simultaneously or one at a time.
- Up to 9 separate jobs.
- Free 30-day trial period.
- Have a true multi-user system with the MULTI-TERMINAL-MONITOR (MTM) option.

★ More cost-effective than buying a second machine.

MULTI-JOB	$159.00
MULTI-TERMINAL-MONITOR	$159.00
MULTI-JOB / MTM PACKAGE	$295.00
ELECTRONIC DISK	$59.00
SPOOL PROGRAM	$24.00
SET MEMORY UTILITY	$24.00

B&L Computer Consultants
7337 Northview, Suite B
Boise, Idaho 83704
(208) 377-8088

VISA MasterCard

ealer inquiries are welcomed. Call or write for a free catalog.

CIRCLE NO. 155 ON READER SERVICE CARD

WATSOFT**PRODUCTS**INC

A subsidiary of the University of Waterloo

Announces

a full screen editor and a family of language interpreters for the IBM Personal Computer

APL
BASIC
COBOL
EDITOR
FORTRAN
PASCAL

For further information contact:

WATSOFT Products Inc.
158 University Ave.
Waterloo, Ontario
N2L 3E9

(519) 886-3700
Telex No.: 06-955458

CIRCLE NO. 242 ON READER SERVICE CARD

Engineering Software

for your PC

- *Power plant and chemical process flowsheet analysis*
- *Pressure drops in pipes and fittings*
- *Thermodynamics of steam, air, fuels, combustion products*
- *Component simulation*
- *Solvers and optimizers*

Call or write for our free catalog

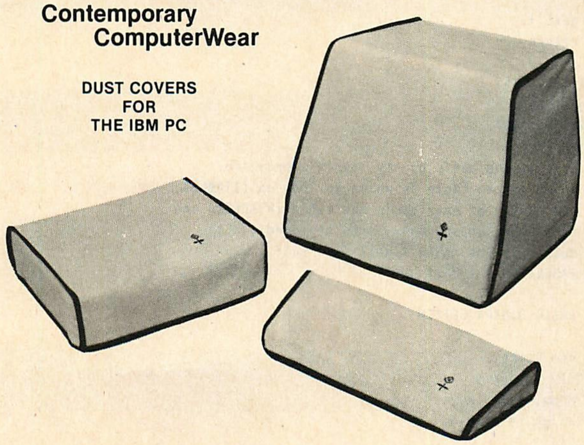
PSI/Systems

Research Park, P.O. Box 3100
Andover, MA 01810
(617) 475-9030

CIRCLE NO. 167 ON READER SERVICE CARD

Contemporary ComputerWear

DUST COVERS FOR THE IBM PC



Attire your IBM PC in style! **ComputerWear** is dust protection with class. Design features include select, woven fabric and embroidered emblem. Order now. Satisfaction guaranteed.

☐ **YES,** PROTECT MY PC IN STYLE! SEND ME: ☐ CHECK

____ Keyboard(s) $16	____ Monitor(s) $18	☐ VISA
____ Drive(s) $18	____ Printer(s) $18	☐ MC
____ 1pc Mon/Drive$36	Add $2.00 Shipping	Total: $______

(CA. Res. add tax)

Print Name: ____________
Address: ____________
City/State: ____________ Zip______
Credit Card #: ____________ Exp.______
Signature: ____________

Contemporary ComputerWear Att: TJ
1320 36th Ave./San Francisco, CA 94122 • 415/564-0506
IBM PC is a trademark of International Business Machines Inc.

CIRCLE NO. 258 ON READER SERVICE CARD

```
'    SEQUENTIAL SEARCH
'
8000 rem Start.
REM CMP   cmp = 0
'
     for i. = 1 to ntable
REM CMP cmp = cmp + 1.
     if (itable(i.) = jkey) then indx = i.: return
     next i.
'
     return
'
'$$$$$$$$$$$$$$
'Subroutine
'$$$$$$$$$$$$$$
'
'    This is the modified sequential search subroutine.
'    Input is the table in which to look up (ITABLE),
'    the number of entries in the table (NTABLE), and
'    the number whose index is to be found (JKEY).
'    Output is the index (INDX) such that
'    TABLE(INDX) = JKEY
'
'    MODIFIED SEQUENTIAL SEARCH
'
8000 rem Start.
REM CMP   cmp = 0
'
     for i. = 11 to ntable+10 step 10
REM CMP     cmp = cmp + 1.
       if (ntable <= i.) then goto 8010
       if (itable(i.) < jkey) then goto 8020
'
'    ITABLE(i.) < jkey; go back and look through last 10 entries.
'
8010   jstart. = i.-10
       for j. = jstart. to i.
REM CMP       cmp = cmp + 1.
         if (itable(j.) = jkey) then indx=j.: return
       next j.
'
8020 next i.
     return
'
'$$$$$$$$$$$$$$
'Subroutine
'$$$$$$$$$$$$$$
'
'    This is the basic binary search subroutine.
'    Input is the table in which to look up (ITABLE),
'    the number of entries in the table (NTABLE), and
'    the number whose index is to be found (JKEY).
'    Output is the index (INDX) such that
'    TABLE(INDX) = JKEY
'
'    BASIC BINARY SEARCH
'
8000 rem Start.
REM CMP   cmp = 0
     ilow. = 1
     ihigh. = ntable
'
8020 rem Loop.
       imid. = (ihigh.+ilow.)\2    'Integer division truncates!
REM CMP     cmp = cmp + 1
       if (itable(imid.) = jkey) then indx   = imid. : return
       if (itable(imid.) > jkey) then ihigh. = imid. - 1
       if (itable(imid.) < jkey) then ilow.  = imid. + 1
     goto 8020
'
     return
'
'$$$$$$$$$$$$$$
'Subroutine
'$$$$$$$$$$$$$$
'
'    This is the assembly language binary search subroutine.
'    Input is the table in which to look up (ITABLE),
'    the number of entries in the table (NTABLE), and
'    the number whose index is to be found (JKEY).
'    Output is the index (INDX) such that
'    ITABLE(INDX) = JKEY
'
'    ASSEMBLY LANGUAGE BINARY SEARCH
'
8000 rem Start.
'
     call binsearch (ITABLE(1), NTABLE, JKEY, INDX)
'
     return
```

LISTING 3 ASSEMBLER BINARY SEARCH SUBROUTINE

```
;
ASM_SEG        SEGMENT PARA PUBLIC
        ASSUME  CS:ASM_SEG
        PUBLIC    BINSEARCH
BINSEARCH PROC FAR
;
;    THIS SUBROUTINE IS TO BE CALLED FROM BASIC OR
;    ANOTHER HIGH-LEVEL PROGRAMMING LANGUAGE.  IT
;    DOES A BINARY SEARCH ON A DATA TABLE,
;    RETURNING THE DESIRED INDEX INTO THE TABLE.
;
;    FORMAT:
;
;         CALL BINSEARCH (itable, ntable, jkey, indx)
;
;    WHERE     itable     IS A TABLE OF INTEGER VALUES
;         ntable     IS THE NUMBER OF ITEMS IN THE TABLE
;         jkey IS AN INTEGER WHOSE LOCATION IN THE TABLE
;         IS DESIRED indx IS THE RESULT.
;         jkey = itable(indx)
;
;    NOTES:
;         ITABLE MUST BE IN ASCENDING ORDER.
;         JKEY MUST BE A VALID KEY, I.E., IT MUST BE
;         CONTAINED IN ITABLE.
;
;    SEE APPENDIX C. OF THE BASIC MANUAL FOR DISCUSSION
;    OF THE BASIC CALLING SEQUENCE.
;
         PUSH BP              ;YES, THIS IS NECESSARY!
         MOV  BP, SP
;
         MOV  SI, [BP]+12     ;ADDRESS OF ITABLE
         MOV  DI, [BP]+10     ;ADDRESS OF NTABLE
         MOV  DX, [DI]        ;NTABLE IN DX
         MOV  DI, [BP]+8      ;ADDRESS OF JKEY
         MOV  AX, [DI]        ;JKEY IN AX
         MOV  DI, [BP]+6      ;ADDRESS OF INDX
         MOV  BP, SI          ;MOVE ADDRESS OF ITABLE TO BP
;
;    INITIALIZE ILOW, IHIGH
;
         MOV  CX, 1                ;CX=ILOW=1
;                             DX=IHIGH=NTABLE
SEARCH_LOOP:
;
         MOV  SI, DX          ;IHIGH INTO SI
         ADD  SI, CX          ;IHIGH+ILOW IN SI
         SHR  SI, 1           ;DIVIDE BY 2 (TRUNCATE): IMID
;
         PUSH SI              ;SAVE IMID, CALC. OFFSET NEXT
         DEC  SI              ;ITABLE(1) IS OFFSET 0
         SHL  SI, 1           ;TWO BYTES PER ENTRY
;
         CMP  [BP][SI], AX    ;COMP. ITABLE(IMID) WITH JKEY
;
         POP  SI              ;RECOVER IMID
;
         JE   CLEAN_UP        ;JKEY = ITABLE(IMID), FOUND IT
         JG   JKEY_SMALLER    ;JKEY < ITABLE(IMID)
;
         MOV  CX, SI          ;JKEY > ITABLE(IMID), NEW ILOW
         INC  CX              ;ILOW = IMID + 1
         JMP  SEARCH_LOOP
JKEY SMALLER:
```

```
-
        MOV  DX, SI          ;NEW IHIGH
        DEC  DX              ;IHIGH = IMID - 1
        JMP  SEARCH_LOOP
;
;   END SEARCH_LOOP
;
CLEAN_UP:
        MOV  [DI], SI        ;STORE INDX
;
        POP  BP              ;RESTORE BP
        RET  8               ;4 ARGUMENTS
;
BINSEARCH ENDP
;
ASM_SEG      ENDS
;
        END
```

LISTING 4 UNASSEMBLY OF BINARY SEARCH SUBROUTINE

```
u 5210 5246
04C5:5210 55         PUSH BP
04C5:5211 8BEC       MOV  BP,SP
04C5:5213 8B760C     MOV  SI,[BP+0C]
04C5:5216 8B7E0A     MOV  DI,[BP+0A]
04C5:5219 8B15       MOV  DX,[DI]
04C5:521B 8B7E08     MOV  DI,[BP+08]
04C5:521E 8B05       MOV  AX,[DI]
04C5:5220 8B7E06     MOV  DI,[BP+06]
04C5:5223 8BEE       MOV  BP,SI
04C5:5225 B90100     MOV  CX,0001
04C5:5228 8BF2       MOV  SI,DX
04C5:522A 03F1       ADD  SI,CX
04C5:522C D1EE       SHR  SI
04C5:522E 56         PUSH SI
04C5:522F 4E         DEC  SI
04C5:5230 D1E6       SHL  SI
04C5:5232 3902       CMP  [BP+SI],AX
04C5:5234 5E         POP  SI
04C5:5235 740C       JZ   5243
04C5:5237 7F05       JG   523E
04C5:5239 8BCE       MOV  CX,SI
04C5:523B 41         INC  CX
04C5:523C EBEA       JMPS 5228
04C5:523E 8BD6       MOV  DX,SI
04C5:5240 4A         DEC  DX
04C5:5241 EBE5       JMPS 5228
04C5:5243 8935       MOV  [DI],SI
04C5:5245 5D         POP  BP
04C5:5246 CA0800     RET  L,0008
-
```

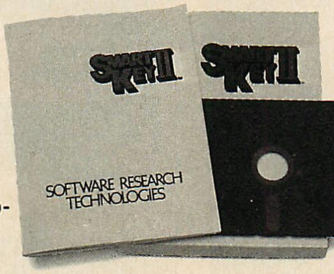
Even the fastest fingers slow down when they struggle with computer commands. And no matter what software you use, you still have to enter the same things over and over again.

That's why you need SmartKey II,™ the software that lets you control your computer with a single keystroke.

Add SmartKey II to WordStar and you can juggle margins and insert "boilerplate" paragraphs or sentences with just one key. Add it to Lotus 1-2-3 and you can concentrate on the numbers instead of the mechanics. You can even tap into Dow Jones with the press of a finger.

Don't type in the same thing twice. SmartKey II can remember commands, words, paragraphs, even entire letters. With SmartKey II, you can work faster and never worry about which keys to press.

It's easy to see why thousands of people all over the world are already using SmartKey II to make their lives easier.



SmartKey II Features:
- Assign more than 3,000 characters to a single key.
- Define over 300 function keys.
- An exclusive "SuperShift" lets every key have up to four meanings.
- Change any keyboard to the Dvorak layout.
- Keys can be redefined while other software is running.
- Key definitions can be saved to disk for later use.
- SmartKey II is invisible to other software.
- SmartKey II runs on the IBM PC and compatibles, the PCjr, the Kaypro II, 4, and 10; plus all other MS-DOS, CP/M, CP/M-86, and CP/M-Plus based personal computers.

SmartKey II is a trademark of FBN Software. WordStar is a trademark of MicroPro, Inc. Lotus 1-2-3 is a trademark of Lotus Development Corporation. CP/M is a trademark of Digital Research, Inc. MS-DOS is a trademark of Microsoft, Inc. PC-DOS is a trademark of IBM.

Best of all, SmartKey II is still only $89.95.

To see how SmartKey II can speed up your work, pick up a copy at your computer dealer. It's the smartest move you will ever make. Software Research Technologies, Inc.,* 3757 Wilshire Boulevard, Suite 211, Los Angeles, CA 90010. (213) 384-5430.

SMART KEY II™

A Stroke Of Genius.

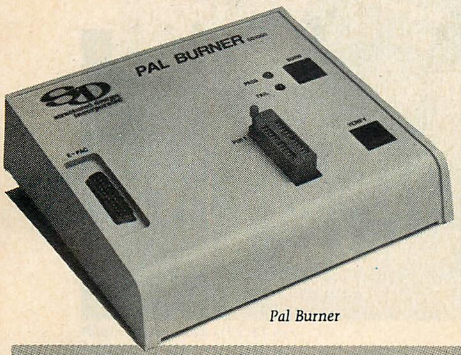
*Formerly Heritage Software, Inc.

CIRCLE NO. 189 ON READER SERVICE CARD

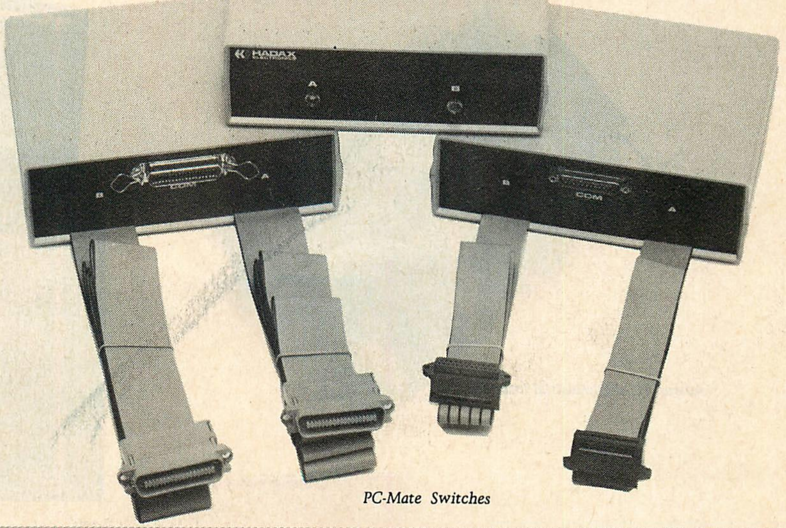
The ten slowest parts of your computer.

TECH RELEASES

Hardware, software, and other developments for the PC



Pal Burner



PC-Mate Switches

IBM has announced a new operating environment derived from Bell Laboratory's UNIX Time-Sharing System. The **IBM Personal Computer Interactive Executive (PC/IX)** is based on INTERACTIVE's IS/3, which is in turn based on UNIX System III, as licensed by AT&T, formerly Western Electric Company. PC/IX is a single-user, multitasking system that features a hierarchical file system; a flexible command language; execution of sequential, asychronous, and background processes; a powerful full-screen editor; a flexible document formatting system; a system to control and track changes in programs and documents; co-residence with PC-DOS on fixed-disk partitions; a high-level programming language (C language) conducive to structured programming; password protection; utilities for file transfer to and from PC-DOS 2.00; and 8087 Match Co-Processor support or emulation.

The hierarchical file system has directories; simple, consistent naming conventions; mountable and demountable file systems and volumes; file-linking across and within directories; a flexible set of directory and file-protection modes; and facilities for creating, accessing, moving, and processing files and directories simply and uniformly.

PC/IX SHELL serves as an interactive command interpreter with high-level programming language constructs. It utilizes, among other features, pipelining, sequential command execution, and program and command execution in the "background" mode.

The INed Editor is a full-screen text editor that features function keys, multiple "windows," overwrite or insert modes, cut-and-paste operations, vertical and horizontal scrolling, "filter" programs, and backup levels. Also included are spelling error detection facilities and text-formatting capabilities.

PC/IX includes a full range of utilities for copying, renaming, deleting, and archiving files. It also includes the Source Code Control System, a collection of programs that manage changes to files.

For structured programming, PC/IX provides a compiler for C, an assembler and relocating loader, a comprehensive run-time library, and a program for checking C source programs for stylistic and portability deviations.

The PC/IX system supports both the IBM PC/XT and the IBM PC with Fixed Disk Expansion. The minimum hardware configuration for the IBM PC/XT consists of: an IBM PC/XT system unit/keyboard, which includes one dual-sided diskette drive, one asynchronous communications support adaptor, and 128 KB storage; two 64 KB memory module kits; one IBM monochrome display and adapter.

The minimum hardware configuration for the IBM PC with Fixed Disk Expansion consists of: an IBM PC system unit/keyboard, which includes one dual-sided diskette drive and 64 KB storage; one IBM PC Expansion Unit with a 10-MB fixed disk drive and fixed disk adapter; three 64-KB memory module kits; and one IBM monochrome display and adapter.

The PC/IX system will have a one-time license charge of $900 and will be available through IBM's National Accounts and National Marketing divisions.

IBM
900 King Street
Rye Brook, NY 10573
914-934-4488

CIRCLE 451 ON READER SERVICE CARD

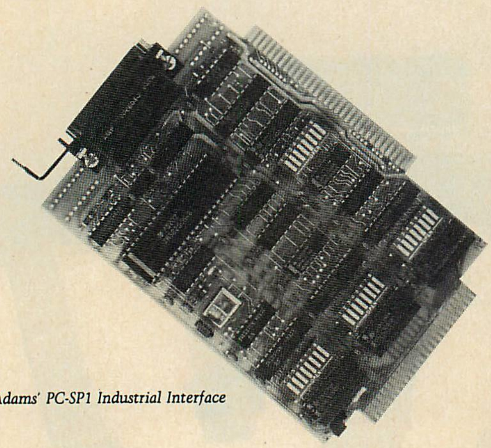
HARDWARE

The **Pal Burner** from **Structured Design** combines full design, simulation and programming aids including menu-driven PALASM in a single product. Design specifications are first checked with the built-in PALASM assembler, which accepts Boolean equations and compiles them into an output to drive the programmer. The PAL Burner programs all currently produced PALs, both 20- and 24-pin devices, from Monolithic Memories National Semiconductor, Texas Instruments, and Advanced Micro Devices. $3,000.

STRUCTURED DESIGN CORPORATION
1700 Wyatt Drive,
Suite #7
Santa Clara, CA 95054
408-988-0725

CIRCLE 487 ON READER SERVICE CARD

PC-Mate data switches from **Hadax Products** include two push-button switches, each with an input panel connector, plus two output connectors mounted on six-foot-long ribbon cables. The switches include one with a 36-pin/32-lead-switched 'Centronics' interface for parallel printers and



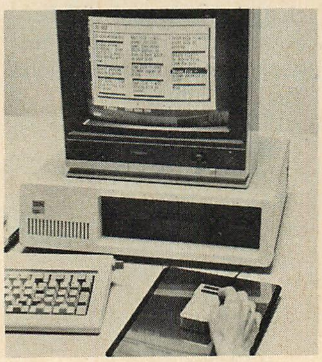
Adams' PC-SP1 Industrial Interface

Ven-Tel's PC Modem Half Card®

one with a 24-lead-switched D-25/RS-232 interface for serial printers. $99.50 each.
HADAX ELECTRONICS
79 Hazel St.
Glen Cover, NY 11542
CIRCLE 482 ON READER SERVICE CARD

Intelligent Technologies has enhanced its **PC Express** Communications Package with **ClusterNET 3270®,** a printed circuit board and SNA software that permit a PC to serve as an IBM 3274 cluster controller linking as many as 12 PCs with a mainframe. The new SNA software also provides a multisession capability, which enables a user whose PC is linked directly with a mainframe to display a variety of IBM 3278/9 terminal screens. Clusternet will be available for delivery in the first quarter of 1984. The new SNA software is available now as a standard feature of the PC Express with SNA package for $1,295.
INTELLIGENT TECHNOLOGIES
151 University Ave.
Palo Alto, CA 94301
415-328-2411
CIRCLE 485 ON READER SERVICE CARD

OptoMouse is a digital mouse from **USI Computer Products** that is available in three software packages: AutoCAD, for drafting and design; VisuAll, which creates a window interface on PCs, XTs, and compatibles; and Blue, a word processor with full-color windowing capabilities. USI has also announced a **modem** for the PC and PCjr. OptoMouse has a software driver that allows its control buttons to be utilized with existing application programs while simplifying application programming. OptoMouse with AutoCad, $1,200; with VisuAll, $485; with Blue, $375. Modem for PCs, PCjrs, $99.
USI COMPUTER PRODUCTS
71 Park Lane
Brisbane, CA 94005
415-468-4900
CIRCLE 486 ON READER SERVICE CARD



The **CmC BUSSter GR** from **Connecticut microComputer Inc.** is a microprocessor-based interface designed to allow any computer with an IEEE-488 interface to communicate with any device that has a standard RS-232 interface. It is programmed through BASIC commands from the host (IEEE-488) computer. It comes assembled and tested with case, power supply, and standard female GPIB and DB-25 connectors. $495.
CONNECTICUT MICROCOMPUTER
36 Del Mar Drive
Brookfield, CT 06804
203-775-4595
CIRCLE 480 ON READER SERVICE CARD

A **PC-SPI Industrial Interface** for the PC and XT is available from **Adams.** The PC-SPI incorporates an RS-232 or RS-422 serial port and three parallel input and output ports on a single board, allowing the user to interface to the PC and XT various peripherals for handling industrial serial and parallel data acquisition and control. $219.
ADAMS
P.O. Box 8330, Station A
Greenville, SC 29604
803-233-4342
CIRCLE 484 ON READER SERVICE CARD

A new version of **The Key** is available from **STAFF** for controlling remote access to a computer system. The Key easily attaches to the serial port of any RS-232 terminal and normally requires no other modification to the hardware. $600 for master and one unit.
STAFF COMPUTER TECHNOLOGY
10457 Roselle St.
San Diego, CA 92121
619-453-0303
CIRCLE 483 ON READER SERVICE CARD

Ven-Tel has announced a PC Modem Half Card® and the 1200 PLUS®, a 1200/300 baud, auto-answer, auto-dial, stand-alone smart modem. The Half Card is a 1200/300 baud auto-answer, auto-dial modem that fits in a small expansion slot on the XT. It includes the software communications package CROSSTALK-XVI. PC Modem Half Card, $549; 1200 PLUS Modem, $495.
VEN-TEL
2342 Walsh Ave.
Santa Clara, CA 95051
408-727-5721
CIRCLE 490 ON READER SERVICE CARD

Legacy Expansion Unit

DASCON-1 is a data acquisition and control interface board for the PC from **MetraByte Corporation.** It plugs directly into one of the expansion slots inside the PC and functions as a high-precision, low-speed data acquisition and control system and can connect directly to your application. $485.

METRABYTE
CORPORATION
254 Tosca Drive
Stoughton, MA 02072

CIRCLE 479 ON READER SERVICE CARD

The first multifacet expansion unit for the PCjr will be introduced in the first quarter of 1984 by **Legacy Technologies Limited**. The **Legacy** expansion bus provides 80 pins, the first 60 of which are identical to the PCjr's. The extra 20 pins can support features such as expanded interrupt capabilities, control and synchronization signals for coprocessors, and specialized I/O functions. Standard on the Legacy will be the power supply, LED status panel, expansion bus (4 slots), and a look-alike cabinet made to ride neatly atop the PCjr. Other options will also be available. $395 for base unit, power expansion, and LED display; $450 for port, disk drive, and floppy disk controller; $849 for full package.

LEGACY
TECHNOLOGIES
LIMITED
1414 "O" Street
Suite 100
Lincoln, NE 68508
402-475-PCJR or 475-7257

CIRCLE 477 ON READER SERVICE CARD

PLAN 3000® from **Nestar** is a mid-range, desktop networking system that allows IBM, IBM-compatible, and Apple personal computers to share information and peripherals. With 10-megabyte capacity and a tape streamer for back-up, the PLAN 3000 File Server can support any combination of PC, XT, Apple II, or Apple III workstations with a maximum of 255 workstations and servers per network. Under $10,000.

NESTAR SYSTEMS, INC.
2585 E. Bayshore Road
Palo Alto, CA 94303
415-493-2223

CIRCLE 488 ON READER SERVICE CARD

Eicon Research Ltd. has introduced **DisCache,** a high-speed, high-performance Winchester disk subsystem. The 10- or 20-megabyte subsystem includes a 5¼-inch hard disk, fast RAM cache memory (up to 256K), a specialized microcomputer, and a newly designed incremental back-up system. $3,000.

EICON RESEARCH
LIMITED
Unit 9, Viking Way
Bar Hill, Cambridge CB3
8EL
ENGLAND
(0954) 81825

CIRCLE 481 ON READER SERVICE CARD

SyQuest Technology has added to its line the **SQ338F Fixed Disk Drive,** a 38-megabyte, half-height Winchester disk drive designed for space-conscious applications requiring high-capacity on-line storage. The unit measures 1.625 × 4.8 × 8 inches. $1,100 in quantities of 1,000.

SYQUEST
TECHNOLOGY
47923 Warm Springs Blvd.
Fremont, CA 94538
415-746-0911

CIRCLE 478 ON READER SERVICE CARD

SOFTWARE

R:base 6000, a multi-user relational DBMS, has been introduced by **Microrim.** It combines the search and sort power, reporting features, and user convenience of Microrim's R:base 4000 with the file organization, management, and security capabilities found in a clustered environment. R:base 6000 can be accessed by several stations at one time while still maintaining database integrity.

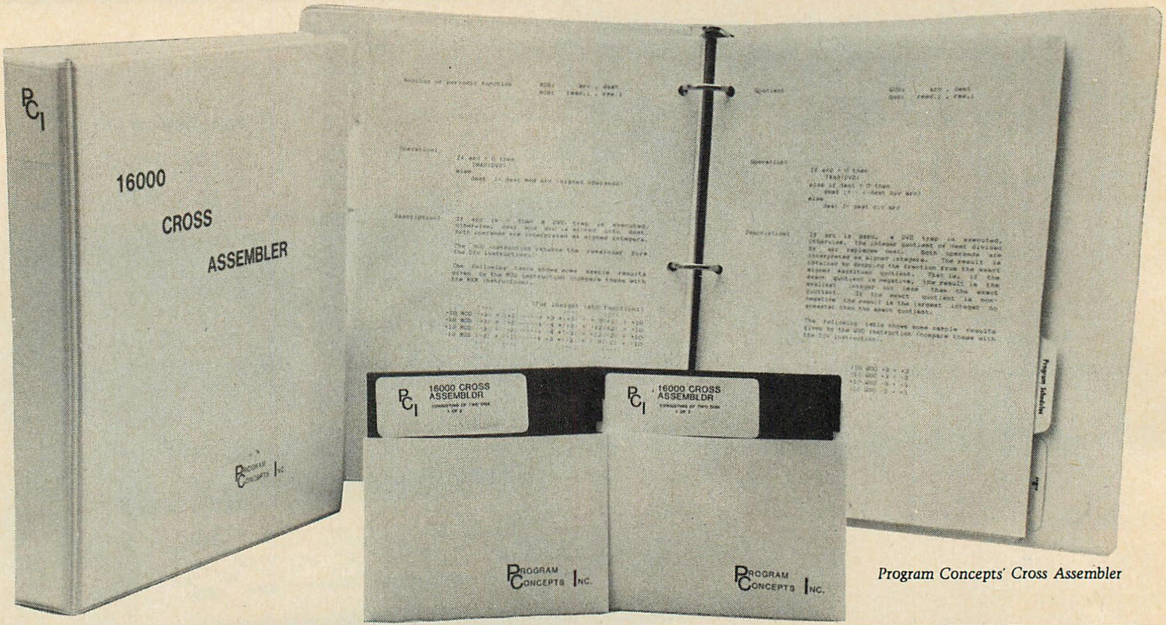
MICRORIM, INC.
1750 112th Avenue, N.E.
Bellevue, WA 98004
206-453-6017

CIRCLE 464 ON READER SERVICE CARD

A new, general-purpose, micro-to-mainframe interface software program from **The Mega Group—Mega-Link/3270**—provides the IBM PC, PC/XT, and PCjr with dial-up access to IBM mainframes, allowing them to function as IBM 3270-type terminals. $2500 for ten copies; $150 per additional copy.

THE MEGA GROUP
2091 Business Center
Drive, Suite 100
Irvine, CA 92715
714-752-9533

CIRCLE 459 ON READER SERVICE CARD



Program Concepts' Cross Assembler

The latest entries in the **XMAC** series of cross assemblers from **Allen Ashley** enable any MS-DOS/PC-DOS system to serve as a development station for the Texas Instruments TMS1000/1100 family; the NEC 7500; the NEC mu-COM-4 microprocessors; the Motorola 6800/01/03/11, 6805, or 6809; the Intel 8085; the Zilog Z8; the RCA 1802/4/6; the Fairchild F8/3870; the National Semiconductor 8070; or the MOS Technology 6502. The software features a macro-assembler, a cross-reference generator, a screen editor, a hex file converter, and off-loading facilities. $150.

ALLEN ASHLEY
395 Sierra Madre Villa
Pasadena, CA 91107
213-793-5748

CIRCLE 470 ON READER SERVICE CARD

Software Arts has announced an upgraded 1.2 version of **TK!Solver** that includes a faster on-line help facility, as well as increased speed in loading, clearing, direct solving, backsolving, and iterative solving. $399.

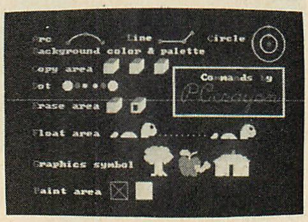
SOFTWARE ARTS, INC.
27 Mica Lane
Wellesley, MA 02181
617-237-4000

CIRCLE 458 ON READER SERVICE CARD

PCcrayon, a multi-use color graphics program for the IBM PC, has been released by **PCsoftware of San Diego.** With this system, the user can change colors and sizes of symbols or move segments around from one place to another. $44.95 to $125.

PCSOFTWARE OF SAN DIEGO
9120 Gramercy Drive
Suite 416
San Diego, CA 92123
619-571-0981

CIRCLE 472 ON READER SERVICE CARD



LETTER EXPRESS, a new software application package from **Software Marketing Group,** allows IBM-PC users to access the Postal Service's Electronic Computer Originated Mail (E-Com). A computer-generated letter along with a list of mailing addresses can be transmitted via modem to the E-Com system, where the message will be merged with the mailing list, typed (up to two pages), inserted, and sent to its destination by first-class mail. Total cost for a single page is 26 cents; 5 cents more for a two-page letter. LETTER EXPRESS guarantees that the user will pass E-Com's certification tests on the first try.

SOFTWARE MARKETING GROUP
6308 Troost
Kansas City, MO 64131
913-791-2720

CIRCLE 461 ON READER SERVICE CARD

Two new versions of **Concurrent CP/M, Digital Research**'s single-user, multi-tasking operating system for personal computers, were recently announced. One of these packages, a generic version of Concurrent CP/M for OEMs who use Intel's 8086 or 8088 microprocessors, can be configured for single-user and multi-user systems and is compatible with PC-DOS. The other version, a retail package that is also intended for use with the IBM PC, includes a window capability. The OEM version will be available at OEM-negotiable prices; the retail version will sell for $150.

DIGITAL RESEARCH
160 Central Avenue
Pacific Grove, CA 93950
408-649-3896

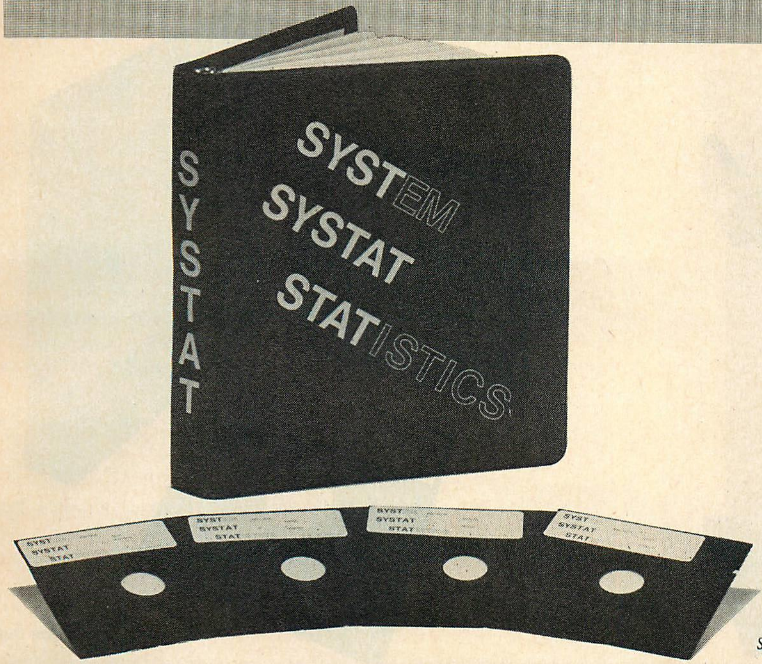
CIRCLE 476 ON READER SERVICE CARD

10-BASE, a relational data base management software package based on the Sequel relational DBMS language originally developed by IBM for mainframe computers, was recently announced by **Fox Research.** The system runs on MS-DOS and requires 192k of RAM. 10-BASE can exchange information with Lotus 1-2-3, WordStar, and other popular software packages. $495.

Also new from Fox Research is **10-NET,** an inexpensive local area network system that can link up PCs, allowing users to share information, explore relationships, make presentations, and prepare business reports. Every plug-in card in 10-Net contains both "user" and "server" control software, which adds flexibility to the networking system. $595.

FOX RESEARCH, INC.
7005 Corporate Way
Dayton, OH 45459
513-443-2238

CIRCLE 469 ON READER SERVICE CARD



SYSTAT

Program Concepts, Inc.'s new **Cross Assembler** for the NS16000 permits the IBM PC to make programs for the new generation 32-bit computers. Written in C per K and R, the package consists of four utility programs: Cross Assembler, Cross Link, Debugger, and Librarian. $595.

PROGRAM CONCEPTS, INC.
P.O. Box 8164
Charlottesville, VA 22901
804-978-1850

CIRCLE 463 ON READER SERVICE CARD

Whitesmiths has introduced a new version of its **Idris UNIX-compatible operating system.** The new version is designed to coexist with MS/PC-DOS and provides for IBM PCs and compatibles the efficient multitasking capabilities of UNIX-style systems.

Also new from Whitesmiths are **compiler packages** for porting applications from the UNIX or Uniplus operating systems to 8080- and 8026-based machines running CP/M or MS/PC-DOS. These new compiler packages make it easier to move UNIX-developed applications to microcomputers.

The new **Whitesmiths Application Software Directory** is also now available. It includes product profiles, hardware requirements, and pricing/ordering information for Idris-compatible software products.

$550—Idris; $175—annual maintenance, including technical telephone support and software updates; $550—C Compiler packages; $700—combined C and Pascal compiler package; $1100—UNIX, Uniplus, and other Whitesmiths cross compiler packages, including compilers for host and target machines.

WHITESMITHS LTD.
97 Lowell Road
Concord, MA 01742
617-369-8499

CIRCLE 455 ON READER SERVICE CARD

Statistics capabilities formerly available only on mainframes are now available for micros, through **SYSTAT,** a comprehensive statistics and data management system from **Systat, Inc.** In addition to the usual univariate and multivariate statistical and graphical procedures, SYSTAT includes more sophisticated routines for probability plots, exploratory data analysis, log-linear analysis of multi-way tables, and factor analysis, as well as the only microcomputer implementations of multidimensional scaling, the multivariate general linear model, and multivariate analysis of variance. $495 for four double-density floppy disks in standard formats.

SYSTAT, INC.
1127 Asbury Avenue
Evanston, IL 60202
312-866-5670

CIRCLE 467 ON READER SERVICE CARD

Phaser Systems, Inc. has announced its **VDAM** (Virtual Diskette Access Method), a software package that completely integrates the mainframe and micro environments by providing users with unlimited mass storage, information exchange, automatic personal computer backup, and audit control. $300.

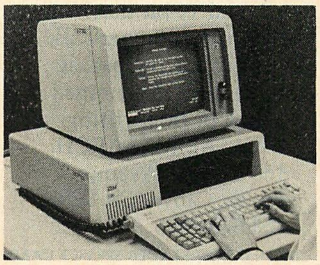
PHASER SYSTEMS, INC.
24 California Street
San Francisco, CA 94111
415-434-3990

CIRCLE 471 ON READER SERVICE CARD

Release 1.1 of **ExecuStation,** a processing service linking PCs to Informatics mainframes, has been announced by **Informatics General Corporation.** New features include automatic log-on to an Informatics data center through Telenet, autodial support for the Hayes Smartmodem, and intelligent terminal capabilities. $200 for users who meet the monthly minimum charge for mainframe processing of $300. $50 per connect hour for file transfer.

INFORMATICS GENERAL CORPORATION
21031 Ventura Boulevard
Woodland Hills, CA 91364
213-887-9040

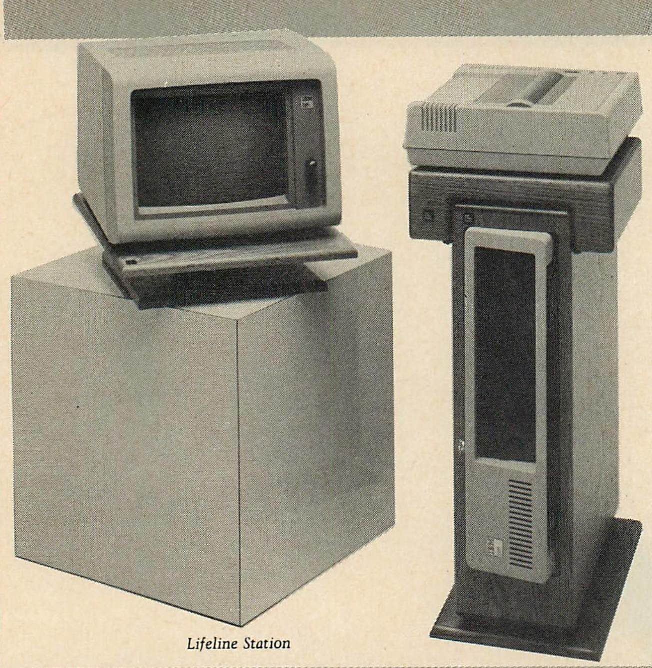
CIRCLE 460 ON READER SERVICE CARD



Information Builders, Inc. announces the availability of **FOCUS,** its nonprocedural database and information management system, for the IBM PC XT/370. FOCUS's VM/CMS is available under the XT/370's announced VM/PC operating system. FOCUS will be licensed as a rider to a user's existing mainframe license—$375/month rental or $8,400 purchase.

INFORMATION BUILDERS, INC.
1250 Broadway
New York, NY 10001
212-736-4433

CIRCLE 454 ON READER SERVICE CARD



Lifeline Station

Omni Resources' Intelligent Disks

Also from **Digital Research** comes **DR Soft/Net,** a high-level CP/M networking software package that is compatible with all local area networking hardware. DR Soft/Net interconnects microcomputers running any version of the Concurrent CP/M operating system. Both 8- and 16-bit versions will be available, allowing 8- and 16-bit machines to run on the same network. The price for the package had not been determined at the time of this writing.

DIGITAL RESEARCH
160 Central Avenue
Pacific Grove, CA 93950
408-649-3896

CIRCLE 475 ON READER SERVICE CARD

OTHER WARES

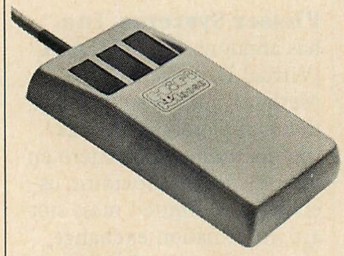
Display Interface Corporation and **Key Tronic Corporation** have announced an agreement authorizing Key Tronic to manufacture and sell Display Interface's **HiFi Mouse.** The agreement also provides for continuing cooperation between the two companies in the development of mouse technology.

DISPLAY INTERFACE CORPORATION
1770 Post Road
Milford, CT 06460
203-877-7661

CIRCLE 432 ON READER SERVICE CARD

KEY TRONIC CORPORATION
Spokane Industrial Park
Spokane, WA 92216
509-928-8000

CIRCLE 431 ON READER SERVICE CARD



Lifeline Information Systems announces the **Lifeline Station,** a three-piece, solid wood, workstation for your IBM PC, XT, IBM Expansion Chassis, or Dec Rainbow. The Lifeline Tower houses your PC vertically beside your desk, a printer stand fits over the Lifeline Tower or can be placed on your desk, and a monitor stand that tilts or swivels completes the set. $79.00 for printer stand, $99.00 for monitor stand, and Lifeline Tower starts at $199.00.

LIFELINE INFORMATION SYSTEMS, INC.
P.O. Box 766
Sandy, Utah 84091
801-566-5340

CIRCLE 445 ON READER SERVICE CARD

A **sales and rental service** of the IBM PC Software Users Group library has been started by the **National Public Domain Software Center.** There are more than 100 programs in this library and all are in the public domain and not copyrighted. They may be freely copied and exchanged. $6.00 for "FLIPPY" disk with a volume on each side. $99.50 for seven days rental of the complete library after receipt, with three days grace for return. Add $7.50 for shipping, handling, and insurance for rentals. Telephone orders may be left on a 24-hr. telephone message machine by leaving credit card information.

NATIONAL PUBLIC DOMAIN SOFTWARE CENTER
1062 Taylor Street
Vista, CA 92083
619-727-1015 (credit cards)
619-941-0925

CIRCLE 438 ON READER SERVICE CARD

A new line of high-quality disks with unique sensors has been introduced by **Omni Resources, Inc. Intelligent Disks** have two sensors that warn of high temperatures and excessive humidity, which may lead to loss of data. $52.50 for a box of ten single-sided, double-density disks.

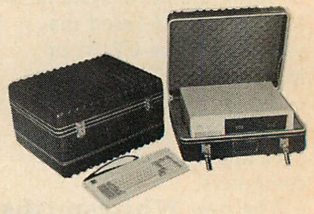
OMNI RESOURCES, INC.
50 Howe Avenue
Millbury, MA 01527
617-865-4451

CIRCLE 426 ON READER SERVICE CARD

Fiberbilt has introduced a new line of fully foam-padded and key-lockable **cases for the IBM PC.** Case #19032 holds the IBM color or monochrome monitor and an IBM or Epson MS-80 printer. Case #19023 holds the CPU and keyboard. Case #19032—$191.00; case #19023—$181.00.

FIBERBILT
601 W. 26th Street
New York, NY 10001
212-675-5820 (in NY)
800-847-4176

CIRCLE 444 ON READER SERVICE CARD



TECH BOOK

A Special Section for Product and Service Listings

ACCESSORIES/SUPPLIES

IBM PC SECURITY
Deter unauthorized persons from opening PC system unit. Kit includes specially designed Clarke screwdriver & companion Kent screws, kit price $18.95. Control multiple PC'S..qty disc & extra screws available. To order send ck or money order:
HOOKER TECH/ETERIA
PO Box 461
Lawrence, NY 11559

HARDWARE/ADD-ON BOARDS

EPROM & MP PROGRAMMER
PC compatible PROM-2000 card can program 2716, 32, 32A, 64, 128, MCM 68764 EPROMS and also 8748/49/51 processors. The software (CP/M-86, MSDOS) can read, verify and program eproms and uses fast programming algorithm. The zero insertion socket is mounted on an external box. The external box for BI-POLAR proms and PAL is also available.
ADVANCED MICROCOMPUTER SYSTEMS, INC.
6802 N.W. 20th Ave.
Ft. Lauderdale, FL 33309
305-975-9515

DT2801 SERIES ANALOG I/O
Plug-in data acquisition boards with 8DI/16SE analog inputs, high or low level programmable gains, 2 analog outputs, 16 lines of digital I/O, DMA, on-board clock, and on-board microprocessor. Optional software subroutines and screw terminal panels.
DATA TRANSLATION INC.
100 Locke Drive
Marlboro, MA 01752
(617) 481-3700

HARDWARE/PERIPHERALS

GRAPHICS ON YOUR NEC
The only complete printer buffer family! Print and compute at the same time. Buffers from 8K to 64K bytes for either parallel or serial printers. Graphics models convert 1-2-3 and MBA graphics output into high quality dot graphics on letter quality printers. Units ungradable to larger buffers and graphics. Proprietary double-buffering. Switch selectable hardware/software protocol. Cables and power supply included. Unique case mounts to side of PC without tools. Priced from $229.

ConvertaBuffer™

VON LEIVENDYKE ENTERPRISES
Silvermine Avenue
Norwalk, Conn. 06850

HARDWARE/DISK DRIVES

8" & 5¼" FLOPPY DRIVES AND DISKETTES
• 8" flexible disk subsystem—Attractive thin-line design stacks above or below system's unit (19½" × 18" × 3½") ANT with complete documentation. Controller board and software available.
I-8480 DD, DS, 2.4 MB $1495
I-8481 SD, DS, 1.2 MB $995
• 5¼" internal disk drives—TANDON or Control Data—DS, 40 ATP1, 320 KB.
TM 100-2 $235 • CDC 9409 $249
5¼" half height internal drives available.
• Control Data diskettes—1240-00 5¼" SS/DD w/write protect notch in hub ring—bx of 10 $22. 1244-00 5¼" DS/DD $35. 1225-00 8" DS/DD wpn $39.50. VISA/MC.

MICROXPRESS

MICROXPRESS
305 S. State College, Suite 135
Anaheim, CA 92806
(714) 632-8512

HARDWARE/STORAGE DEVICES

WINCHESTER SUBSYSTEMS
Tallgrass Technologies offers a family of Winchester HardFiles and streaming tape by file backup for the IBM PC. With formatted capacities from 6.25MB to 70MB, Tallgrass has a HardFile System to fit the most stringent of data processing applications. From $2995 including integral tape backup.
STEVE VOLK/V.P. MARKETING
Tallgrass Technologies Corporation
11667 West 90th
Overland Park, KS 66214
(913) 492-6002

PUBLICATIONS

DYNAMIC DUO RETURNS!
Two new disk magazines for the IBM-PC-PC FIRING LINE (for programmers) and PC UNDERGROUND (for non-tech folk) are available now. Send a self-addressed stamped disk mailer and two formatted DS/DD disks for your free copy.
ABCOMPUTING
P.O. Box 5503
North Hollywood, CA 91616-5503
(818) 509-9002

MAILING LISTS

IBM MAILING LISTS
Over 75,000 names of IBM personal computer owners (counts increase daily) available for rental on labels or magnetic tape. Total 550,000 including other brands.
IRV BRECHNER
TARGETED MARKETING, INC.
Box 453
Livingston, NJ 07039
(201) 731-4382

PROFILES OF RETAIL STORES & USERS
CI can provide names & addresses of: a) over 2,200 retail computer stores surveyed via the telephone, b) new personal computer users (e.g. IBM, Apple, Osborne, etc.). Both can be tailored & used for mailing lists, telemarketing or direct sales. Call (619) 450-1667.

COMPUTER INTELLIGENCE CORP.
3344 North Torrey Pine Ct.
La Jolla, CA 92037

SOFTWARE/ACCOUNTING

SOLOMON III INTEGRATED ACCOUNTING
Suitable for large and small businesses, banks, and accounting firms. Unsurpassed in flexibility, power, and ease of use. Includes General Ledger, Accounts Payable, Accounts Receivable, Payroll, Fixed Assets, Inventory, Purchasing, Order Entry, Sales analysis, Job Costing, and Mail List integrated around the MDBS III database management system. Features graphically displayed screens, on-line help text, and more accounting features than any other microcomputer accounting system. Complete documentation. Modules start at $595. Demo available. See our ad this issue.

COMPUTECH GROUP INCORPORATED
Lee Blvd., Box 271
Frazer, PA 19355
(215) SOLOMON or 644-3344

SOFTWARE/BUSINESS PROJECT MANAGEMENT

PRIMAVERA PROJECT PLANNER™
Project planning, management and control. Uses critical path method. Offers interactive control of project activities, schedules, costs and resources. Up to 10,000 activities in any work breakdown structure. Tabular and graphic reports support earned value analysis. Call or write for brochure and sample reports.
PRIMAVERA SYSTEMS, INC.
29 Bala Avenue, Suite 224
Bala Cynwyd, PA 19004
(215) 667-8600

SOFTWARE/COMMUNICATIONS

VT 100/102/52 TERMINAL
PC100 turns your IBM type PC into a VT100/102/52 terminal. With PC 100 you can use DEC applications such as EDT, KED, WORD 11, & vi. PC 102 adds VT102 printer support & text file transfer to the standard VT 100 features. PC 102—SV offers 132 column support. Full VT keyboard emulation, easy to use SET-UP screens. Call or write for more information. From $89.
GENERAL MICROSYSTEMS
7525 Mitchell Rd., Suite 101
Minneapolis, MN 55344
(612) 937-9194

RATES AND INFORMATION

Listings are grouped by category and consist of a bold lead line (23 characters maximum), 7 lines of ad copy (45 characters per line), plus 4 lines of company name, address and telephone number.

Listings are available only on a 3 issue basis at $60. per issue ($180. total). Copy will have a set format and remain the same for all 3 months. Enhance the appearance of your ad by including your Logo at an additional cost of $25. per issue ($75. minimum extra charge). Pre-payment is required by check, money order, or American Express, Diners Club, MasterCard, Visa credit cards. Closing Date: 1st of 2nd month preceding cover date.

Send copy and remittance to PC TECH JOURNAL, TECH BOOK, 12th FLOOR, 1 PARK AVENUE, NEW YORK, NY 10016. Call (212) 725-4215 for additional information or assistance.

TECH BOOK

Software/Compiler

C86—"C" FOR PROFESSIONALS

"C" brings great power, control and conciseness to programming. C86 brings FULL UNIX syntax (portability), source to a rich library (gives you control), no royalties, PCDOS & CPM-86 support, and overlays. Ask a friend—we are ahead. Development will keep us ahead. Hotline support is solid. $395. Call your Dealer or CI.
COMPUTER INNOVATIONS, INC.
10 Mechanic Street, Suite J-113
Red Bank, NJ 07701
(201) 530-0995

THE PROGRAMMER'S SOURCE

COMPARE products. Find what's best. We serve programmers of MSDOS and CPM systems. We cover all compilers (like 15 PCDOS C compilers), editors, utilities, operating systems, communications programs, addon hardware, key books and supplies. Call for answers, Referral Service, a catalog, literature, prices.
THE PROGRAMMER'S SHOP
135 Main Street, Suite 3-D
Maynard, MA 01754
(800) 421-8006, (617) 897-4750

DESMET C—$109

Full K&R C compiler, assembler, linker, librarian, full-screen editor and example software. Both 8087 and floating point libraries. OUTSTANDING PRICE/PERFORMANCE. Rated 1st or 2nd in August '83 BYTE benchmarks. No royalties on generated code. C Ware newsletter. Unlimited updates at $20 each. PC-DOS (Ver 1.1 & 2.0), generic MS-DOS and CP/M-86 support. $109 for complete package, shipping included.

C WARE CORPORATION

C WARE CORPORATION—MAIL ORDER DEPT.
P.O. Box 710097
San Jose, CA 95171-0097

C-SYSTEMS•COMPILER•WINDOW

Working in support of the c-systems full C Compiler, the c-window provides source statement execution and breakpoints; variables may be displayed/altered using C expressions; displays source code as program is tested. A must for all C programmers. 128K, 2DD required. C Compiler $195; c-window $195.
C-SYSTEMS
P.O. Box 3253
Fullerton, CA 92634
(714) 637-5362

Software/Data Management

T.I.M. IV

A data base management system that generates and summarizes files, mailing lists, and employee records. The data can be accessed and searched in a number of ways. T.I.M. provides help screens and function key legends as user aids. Capacities include 32,767 records per file, 40 fields per record, 60 char/field. (List Price: $500.)
INNOVATIVE SOFTWARE, INC.
9300 W. 110th St. #380
Overland Park, KS 66210
(913) 388-1089

Software/Development/Tools

LITTLE KENNY'S TOOL

WRITE BASIC programs up to ten times faster and develop them with efficient, clean, well structured code. Little Kenny's Tool also includes the ability to generate data entry screens and reports. And a b+tree filing system is included for fast, sorted record filing and retrieval. $116.
KENNETH R. WALETZKI
P.O. Box 992
Rosemead, CA 91770
(213) 285-3077

ACTIVE TRACE

A debugging & learning tool that lets you see how your basic program works. As your program runs, active trace shows you your variables & their current value & line number. (Dynamic Symbolic Debugger). Output to screen, printer or disk. No need to single step. Complete XREF mapping. Review in BYTE 4/83, p. 334, & Microcomputing 12/83, p. 22. $79.95. Call toll free for info, (800) 358-9120, U.S.; (800) 862-4948, CA.
AWARECO
P.O. Box 695
Gualala, CA 95445
(707) 884-4019

PASCAL COMPILER ASSIST

WITH IBM PASCAL and DOS 2. Compiler Assist Program (CAP) sets the ERRORLEVEL for BAT files to check return codes from PAS1 and PAS2. Allows completely automated compiles through the linkedit step. No more human intervention—let CAP do the monitoring for you! Tremendous productivity tool. BAT files included. $35.

SLICK SOFTWARE

P.O. Box 641
Harrisburg, NC 28075
(704) 455-5927

FREE FREE FREE

BASIC AIDS FACT SHEETS and our guide titled "MAKE YOUR PC PROFITABLE" are sent free to persons who request them. Learn more about developing Structured Programs in BASIC. This new release of BASIC AIDS is the most powerful program DEVELOPMENT and DOCUMENTATION tool available!
TULSA COMPUTER CONSORTIUM
P.O. Box 707, 1004 North Dogwood
Owasso, OK 74055
(918) 747-0151

COBOL SUBROUTINES

Give IBM's COBOL the functions that were left out. Use all keys including ALT, Ctrl, all 40 function keys, PgUp, PgDn, INS, DE1, Home, End, etc. Rename/Delete files. Change the default disk. Plus Rel 2.0 functions: make/change/remove directories, get volume id's, get/change a file's mode. More. Easy to use. $49.95.
Sun*Soft
PO Box 4277
Ft. Lauderdale, FL 33338
(305) 566-8827

Software/Engineering

PELADA TEXT ENGINEER

Engineering and Scientific Word processing. Technical symbols, special characters, super/subscripts, italics, underlining, bolding. With graphics hardware you can design any symbols you want, up to 384 different characters. All characters visible on screen and printable. Easy English commands. $250.
PELADA INFORMATICA INC.
562 Johnson Street
Kingston, Canada K7L-2A1
(613) 549-1747

ELECTRONIC CIRCUIT ANALYSIS

ECA is a high performance simulator for analog electronic circuits. Features include full editing, worst case, sensitivity analysis, ability to sweep components, 64 nodes. It is easy to use and very fast. Price $150. For all MS-DOS, CP/M, CP/M-86 computers.
TATUM LABS
P.O. Box 698
Sandy Hook, CT 06482
(203) 426-2184

Software/Financial

CALCULATOR EMULATOR

Now you can make quick calculations with the ease of an adding machine or print calculator on your IBM PC. Includes programmable formats for scientific notation, fixed or floating point, auto decimal entries, PLUS optional title line and labelling of entries. For home or business. Dealer inquiries welcomed. $39.95. MC/VISA/CHECK/MO.
SPECIALTY DESIGNS
7739 El Pensador
Dallas, TX 75248
(214) 960-2455

$CASHFLO/$CASHSLO

Track home/business income/expenses by day, month, qtr. or yr. Easy to use menus. Create up to 64 income, expense, check, cash, savings or credit accts. Enter up to 700 single or split transactions/mo. Check sequence search. Tax & check reconcile flags. 2DD/HD, 128K, DOS2.0. $195. CHECK/M.O. Dealer inquiries welcomed.
INTERNATIONAL SOFTWARE CONSULTANTS
P.O. Box 18218
San Jose, CA 95158
(408) 289-8921

Software/General

PIGLATIN TRANSLATOR

PIGLAT will translate an ASCII text into piglatin! The full source code in the language of C (with notes and remarks) is included. The executable file runs under PC-DOS (except version 2.0) or you can use the source with any C and operating system configuration. Entertaining as well as interesting. Prepaid $29.95.
WESTCOMP
23441 Golden Springs, Suite 300
Diamond Bar, CA 91765

Software/Graphics

VERSAWRITER

Complete drawing system for IBM PC includes digitizer drawing board and software—ready to use. Use digitizer to trace drawings to the computer screen. Features include draw & dot mode, fill & airbrush in 100 colors, add text, save and recall picture, dump graphics to printer, make circles, boxes, pie charts & bargraphs, create your own brush palette and more. $299.00.
VERSA COMPUTING, INC.
3541 Old Conejo Road, Suite 104
Newbury, CA 91320-2178
(805) 498-1956

35mm SLIDE FROM YOUR PC

COMPUTER SLIDE EXPRESS converts graphic files produced on the IBM PC into brilliant 35mm color slides with color resolution 400% better than your monitor. Leave your printouts behind. Project yourself into the technology of tomorrow. Use high resolution color slides at your next presentation. COMPUTER SLIDE EXPRESS. $6/slide, $30 minimum order.
VISUAL HORIZONS
180 Metro Park
Rochester, NY 14623
(716) 424-5300

PLOT-10 GRAPHICS EMULATOR

PC-PLOT is a complete emulation of the TEKTRONIX model 4010 graphics terminal. The basic packages range from $40 to $75 plus optional plotter driver software. Write or call for a technical brochure.
MICROPLOT SYSTEMS CO.
1897 Red Fern Drive
Columbus, OH 43229
(614) 882-4786

TECH BOOK

FORTRAN GRAPHICS PACK
60 MicroSoft FORTRAN/Pascal (1.0 or 3.1) SuperSoft FORTRAN callable subroutines. Professional graphics power backed by a 90 page user manual, filled with examples and designed to be helpful to every level of user. Five packages in one! General utility, 2-D interactive, 2-D plots, 3-D plots, and solid models (hidden line removal!) $95.
MICROCOMPATIBLES
11443 Oak Leaf Drive
Silver Spring, MD 20901
(301) 593-0683

TECHTRONICS 4010 EMULATION
High resolution screen/printer graphics for the IBM PC. Full interactive capabilities with file transfer and cross-hair control. Utilizes PLOT 10, ISSCOS DISSPLA, TEL-A-GRAF and SAS protocols. OFF-line review of graphics output. This emulation software is easy to use with high quality graphics at an affordable price. $300.00.
Technological Systems Group
3852 Foxford Drive
Atlanta, GA 30340
(404) 939-3526

SOFTWARE/LANGUAGES

ADA ON A MICRO
JANUS/ADA is an exceptional programming language for software development. Being a subset of the Ada programming language JANUS/ADA implements those features of Ada which are directly needed by professional programmers. These features include separate compilation, user defined data types and helpful program bugging.
R.R. SOFTWARE
P.O. Box 1512
Madison WI 53701
(608) 244-6436

SOFTWARE/MAILING PROGRAMS

PONY EXPRESS XL
Designed for commercial mailings or extra large lists. Super fast machine code multi sort. Carrier route sort. Prints labels: 1-5 across, single line format, or user-defined format. Also CHESHIRE FORMAT. Unlimited files. Multi file merge. Records can be selected by user-defined codes, specific zips or zip ranges. Phone list option. Unique correction & deletion routines for updating. Instant access to all records. Auto repeat for ease of record entry. Interfaces w/most word processors. $238.
COMPUTECH
975 Forest Ave.
Lakewood, NJ 08701
(201) 364-3005

SOFTWARE/SORT

OPT-TECH SORT/MERGE
Extremely fast Sort/Merge program for the IBM-PC. Can sort or merge multiple files containing fixed or variable length records. Run as a DOS command or call as a subroutine, plus many other features. Now also sorts dBASE II files! Compare before you buy any other. Write or call for more info. $99.
OPT-TECH DATA PROCESSING
P.O. Box 2167
Humble, TX 77347
(713) 454-7428

SOFTWARE/UTILITIES

COPY PROTECTION
SLK/F places an assembled or compiled program on a diskette with 4 different copy-resistant features in such a way that it runs normally, but cannot be copied by backup programs such as COPYPC. The rest of the diskette is available as normal, and DOS may be added. Price $150.

OLIVE BRANCH SOFTWARE

OLIVE BRANCH SOFTWARE
1715 Olive Street
Santa Barbara, CA 93101
(805) 569-1682

IBM-PC COBOL PROGRAMMERS
Screen coding is fun, fast and accurate with Qwik-Screen Builder. "Paint" screen layout and create true COBOL source code for screen section in minutes. 100+ pages of documentation. Satisfaction guaranteed or money back. Needs DOS 1.1/2.0, two DS/DD drives, monochrome display and the COBRUN. EXE runtime module. $50/Visa/MC.
THE COMPUTER WORKS
P.O. Box 1371
Coeur d'Alene, Idaho 83814
(208) 667-6152

THE FORMS DESIGNER™
Attention IBM Pascal, FORTRAN, and Assembly users! Save time in designing formatted screen I/O. Interactive Forms Editor allows you to draw lines and boxes, define fields, and edit text. Access forms or read keyboard entry by writing only one line of code. Provides sequential data retrieval and storage. Requires 128K RAM. Only $275 complete. Demo and manual $35. Call or write:
BIT SOFTWARE
PO Box 619
Milpitas, CA 95035
(408) 262-1054

DISK MECHANIC
THE ULTIMATE Floppy Disk Backup & Repair—Utility. Can back up ALL software protected disks written on the IBM PC. Works manually or automatically. Restores deleted files. Files or sectors can be searched, examined & changed. Checks disk drive speed. Req. IBM PC, DOS 1.1/2.0, 128K mem. & 2 floppy drives. $73 Ppd. USA.
MLI MICROSYSTEMS
PO Box 825, Dept. TB
Framingham, MA 01701 USA
(617) 926-2055 for info MC/VISA.

TOOLS FOR PROGRAMMERS
• FILE TOOLKIT: COMPARE lists file differences: COMPRESS shrinks files; ENCRYPT secures valuable data, HEXDUMP lists files in hex and ASCII.
• PASCAL TOOLKIT. PXREF prints program cross-references; PNEAT prettyprints programs; PCASE highlights variables. Just $29.95 each, both for $49.95. VISA/MC accepted.
SEVEN VALLEYS SOFTWARE
P.O. Box 99
Glen Rock, PA 17327
(717) 235-5470

KEY-MASTER FREE TRIAL
Redefine IBM-PC keyboard. Assign up to 600 characters to a key. Create fill-in-the-blanks command strings. Includes KEY-MASTER Profile Editor. New Profiles easily loaded. One keystroke turns KEY-MASTER on or off. Send only $6.00 now for disk. Send $20.00 later, only if you like KEY-MASTER.
ADAMS ENTERPRISES
Rt. 4 Box 23
Shelbyville, KY 40065

COPY PROTECTION BY MSD
The ULTIMATE diskette copy protection system designed by software developers and publishers. A variety of protection methods are available to suit your protection requirements. All IBM PC and XT environments are supported which includes. DOS, BASIC (compiled and interpretive). Pascal, P-System and stand-a-lone systems. Customized systems also available. Requires: 64K, two disk drives.
MICRO-SOFTWARE DEVELOPERS, INC.
214½ West Main Street
St. Charles, Illinois 60174
(312) 377-5151

THE PROWRITER UTILITIES
COMPLETE CONTROL OF YOUR C. Itoh 8510/1550 and NEC 8023. DOS 1.1 + 2.0 compatible for $44.95. PROCNTRL—On-Line Keyboard Control of Printer PROEPSIM—Epson Graftrax Simulator (eg 1-2-3) PROSCR—Text/Graphics Screen Dump, 2 sizes. PROSTALL—Wordstar 3.2 + 3.3 Installation PROSET—Menu Driven Printer. Setup + 2 more.
COURTRIN ENTERPRISES
P.O. Box 231190
San Diego, CA 92123
(619) 569-8308

CONVERTS ANY CPM TO DOS
CROSSDATA CONVERTS ANY DATA/TEXT FILE format from CPMxx to MS/PC-DOS, CPMxx to CPMxx, & MS/PC-DOS to CPMxx. Cross data runs on IBM-PC or comparable computer using MS/DOS 2.0. Cross data is a self-contained program. It does not require any additional hardware or software. To order send $99 check or money order to:

AWARD SOFTWARE, INC.

AWARD SOFTWARE, INC.
236 North Santa Cruz Ave.
Los Gatos, CA 95030
(408) 395-2773

SOFTWARE/WORD PROCESSING

VOLKSWRITER INT'L EDITION™
For IBM-PC and compatibles. On screen tutorials, full-screen editing, on-screen formatting and standard ASCII/DOS files. Permits writing in German, French, English, Spanish & Italian and generation of scientific, mathematical & engineering characters. Multiple keyboard arrangements and communications possible. $225.00.
LIFETREE SOFTWARE, INC.
411 Pacific St., Suite 315
Monterey, CA 93940
(408) 373-4718

PELADA TEXT ENGINEER
Word processing for scientific and multilingual text. All IBM characters. Super/subscripts, italics, underlining, bolding. With graphics hardware, you can use any symbols and any alphabet, up to 384 characters. All symbols visible on screen and printable. Easy English commands. $250.
PELADA INFORMATICS INC.
562 Johnson Street
Kingston, Canada K7L-2A1
(613) 549-1747

USER GROUPS

PASCAL NEWS
Pascal News is the official but informal publication of the Pascal Users' Group (PUG). PUG promotes the use of, and the ideas behind, the programming language Pascal. Pascal News has seven years of back issues at present and documents source programs written in Pascal. Four issues per year—$25.
PASCAL NEWS
2903 Huntington Rd.
Cleveland, OH 44120

TECH MART

PCDISKID allows you to print sorted directory listings in four formats; add, change, or delete volume labels; and recover inadvertently erased files. PCDISKID requires DOS 2.0 and is available for $30.

PC FILECOMPARE compares the contents of two files and displays the differences in record format. Displayed data can optionally be changed. Compares files of different lengths, no limitations as to the number of non-matching characters. PC FILECOMPARE requires DOS 1.1 or higher and is available for $50.

Both programs present the user with an easy to use screen menu and work on floppy or hard disk.

VISA/MASTERCARD

ErgoSoft

P.O. Box 454, Oakhurst, NJ 07755
(201) 493-8352

CIRCLE 281 ON READER SERVICE CARD

ICs PROMPT DELIVERY!!!

SAME DAY SHIPPING (USUALLY)

64K DYNAMIC RAM ICs
200 ns @ $5.97
150 ns @ $6.09

These are the ICs you need to expand your IBM PC memory. Install these yourself in the motherboard or an expansion board. Our prices are often half that of retail stores. Our quality is second to none.

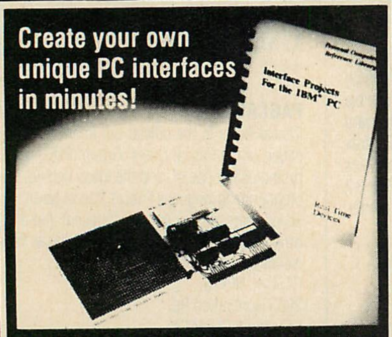
We sell other memory ICs.
27128, 2764, 6264, 256K, etc.

MasterCard VISA or UPS CASH COD
Factory New, Prime Parts µP∞
MICROPROCESSORS UNLIMITED
24,000 South Peoria Ave.
BEGGS, OK 74421
(918) 267-4961

Prices shown above are for Jan. 13, 1984

Please call for current & volume prices. Prices subject to change. Please expect higher prices on some parts due to world wide shortages. Shipping and Insurance extra. Cash discount prices shown. Federal Express Standard Air @ $5.99! Orders received by 6 PM CST can be delivered to you by the next morning.

Create your own unique PC interfaces in minutes!



The PD100 allows rapid development of specialized PC interfaces. It features a buffered data bus, switch-selectable address decoder, and a large prototype area (up to 40 IC sockets). The 116-page manual covers basic interfacing concepts and details implementing A/D, D/A converters, I/O ports, connection of transducers and dozens of useful circuits.

PD100 w/manual - $99
plus $3.50 P&H
Manual only - $20 Postpaid
PA residents add 6%
Check, MasterCard, or VISA

REAL TIME DEVICES
1930 Park Forest Ave.
P.O. Box 906
State College, PA 16804
(814) 234-8087

How to use computers to teach math!

COMPUTERS IN MATHEMATICS:

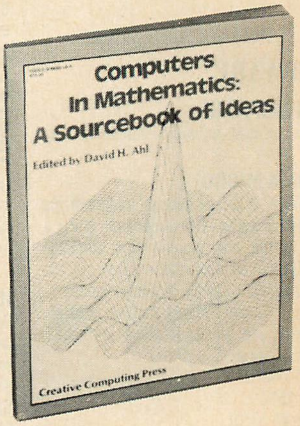
A SOURCE BOOK OF IDEAS

Now that more and more math departments have access to a microcomputer, the problem becomes: How to use the computer effectively as a teaching aid?

Here's where COMPUTERS IN MATHEMATICS can help you. This book of reprints from *Creative Computing* magazine is a goldmine of learning ideas, problem-solving strategies, programming hints, puzzles, brain teasers, and much more!

COMPUTERS IN MATHEMATICS covers nine subject areas:

- Computer literacy and computers in society.
- Thinking strategies and how to solve problems.
- Computer simulations and how to write them.
- Probability.
- Mathematical miscellany, from circular functions to differential equations.
- Art and graphics and their relation to mathematics.
- Computer Assisted Instruction (CAI).
- Programming style.
- Puzzles, problems and programming ideas.



8¼" x 11", softcover, hundreds of black-and-white diagrams and illustrations.

In all, COMPUTERS IN MATHEMATICS contains 77 fascinating articles, over 200 problems for assignment, and nearly 100 programs. Edited and with a preface by David H. Ahl, Editor-in-Chief of *Creative Computing*, this immensely practical volume is an invaluable classroom tool for teachers and students of all grades.

USE THE COUPON TO ORDER YOUR COPY TODAY!

CREATIVE COMPUTING PRESS Dept. NC6B 39 East Hanover Avenue
Morris Plains, New Jersey 07950

Please send me ________ copies of COMPUTERS IN MATHEMATICS: A Sourcebook of Ideas, at $15.95, plus $2.00 postage & handling,* each. #12D

CHECK ONE: ☐ **Payment enclosed $**________ (CA, NJ and NY STATE residents add applicable sales tax.)

☐ **Charge my:** ☐ American Express ☐ MasterCard ☐ Visa

Card No.________ Expire Date________

Signature________

Mr./Mrs./Ms.________
(please print full name)

Address________ Apt.____

City________ State________ Zip________

*Outside USA add $3.00 per order.

For faster service, call **TOLL FREE 800-631-8112.** (In NJ call 201-540-0445.)

INDEX TO ADVERTISERS

PC TECH JOURNAL PRODUCT INDEX

How to teach your kids about the IBM PC.

Computers for Kids teaches children age 8 and older to write their own programs in less than an hour—without the necessity for previous knowledge of algebra, variables, or computers. And there's a special section that keeps parents and teachers on the same successful command path.

Starting off with an easy-to-understand explanation of how to use the IBM PC, your kids will progress quickly to flow charts and simple print programs...to loops, graphics, and other programming concepts that show the young user how to make the PC do exactly what he wants—in non-technical language that makes life easier for both of you.

So do your children a favor.

Do yourself a favor. Order the PC edition of *Computers for Kids* today!

Creative Computing Press
Dept. NP1C, 39 East Hanover Avenue
Morris Plains, NJ 07950

Send me __________ *Computers for Kids*, PC edition, at \$5.95 plus \$1.50 postage and handling each. #12K

☐ **PAYMENT ENCLOSED** \$________
Residents of CA, NJ and NY State add applicable sales tax. Outside USA add \$3.00 per order.

☐ **CHARGE MY:** (Charge and phone order \$10 minimum)
☐ American Express
☐ MasterCard ☐ Visa

Card No.__________ Exp. Date______

Signature__________________

Mr./Mrs./Ms.__________________
(please print full name)

Address______________ Apt.____

City________ State_____ Zip______

Also available at your local bookstore or computer store.

☐ Send me a FREE *Creative Computing* Catalog.

Presenting XyWrite II-*plus*

Word processing—and *more*—when there's no time to waste.

When you want the fastest word processing program, demand XyWrite II. When you want the fastest word processing program with mailmerge and completely redefinable keyboard, insist on XyWrite II-*plus*, new from XyQuest.

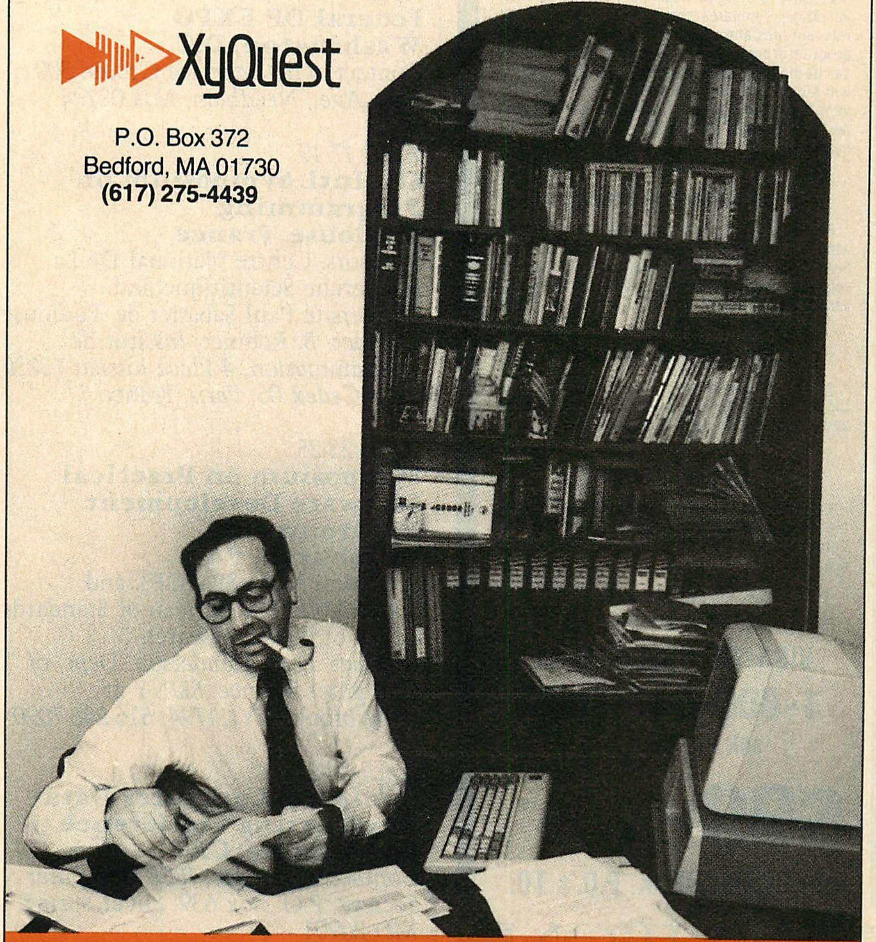
In XyWrite II-*plus* you can insert names and addresses, and other fields from lists created in XyWrite and other programs, and you can assign commands and text to single keystrokes.

For just \$7 we'll rush you a XyWrite II demo disk, money back if you decide to buy. If you really have no time to waste, send \$195 for XyWrite II or \$295 for XyWrite II-*plus*. Add \$5 shipping and handling. MasterCard and Visa welcome.

Split screen • Simultaneous multiple file access • Horizontal scroll • Column moves • What-you-see-is-what-you-print • Indexing • Unlimited footnotes, end-notes • Headers/footers • Boilerplating • Foreign language characters • Background printing • On-line "help" • Executes other programs directly • Extensive printer support • Reference and tutorials • and more . . .

CIRCLE NO. 224 ON READER SERVICE CARD

XyQuest
P.O. Box 372
Bedford, MA 01730
(617) 275-4439



XyWrite II-*plus*

XyWrite II-*plus* is designed for the IBM PC and compatibles.

XyQuest XyQuest

TECH CALENDAR

APRIL

April 5-7
COMDEX/Winter '84
Los Angeles, CA
Contact: The Interface Group, 300 First Ave., Needham, MA 02194

April 9-12
TEEE Infocom 84
San Francisco, CA
Sponsors: IEEE-CS, IEEE Communications Society
Contact: Judy Estrin, Bridge Communications, 10401 Bubb Road, Cupertino, CA 95014, 408-446-2981

April 17-19
Federal DP EXPO
Washington, D.C.
Contact: The Interface Group, 300 First Ave., Needham, MA 02194

April 17-19
6th Intl. Symposium on Programming
Toulouse, France
Sponsors: Centre National De La Recherche Scientifique and Université Paul Sabatier de Toulouse
Contact: B. Robinet, Institut de Programmation, 4 Place Jussieu 75230 Paris Cedex 05, Paris, France

April 23-25
Symposium on Practical Software Development Environments
Pittsburgh, PA
Sponsors: ACM SIGSOFT and SIGPLAN, Natl. Bureau of Standards, Office of Naval Research
Contact: Peter Henderson, Dept. of Computer Science, SUNY at Stonybrook, NY 11794, 516-246-7090

April 24-27
Compdec, Computer Data Engineering Conference
Los Angeles, CA
Contact: Compdec, IEEE Computer Society, P.O. Box 639, Silver Spring, MD 20901

April 26-28
Great Southern Computer Show
Columbia, SC
Contact: Great Southern Computer Shows, P.O. Box 655, Jacksonville, FL 32201, 904-356-1044

April 30-May 2
1984 Symposium on Security and Privacy
Oakland, CA
Sponsor: Technical Committee on Security and Privacy, IEEE
Contact: Peter S. Tasker, The Mitre Corp., MS B325, Bedford, MA 01730

MAY

May 13-17
Computer Graphics 84, the National Computer Graphics Association's Fifth Annual Conference and Exposition
Anaheim, CA
Contact: Education Coordinator, National Computer Graphics Assn., 8401 Arlington Blvd., Suite 601, Fairfax, VA 22031

May 14-18
Fourth International Conference on Distributed Computing Systems
San Francisco, CA
Sponsor: IEEE Computer Society
Contact: IEEE Computer Society, 1109 Spring St., Suite 300, Silver Spring, MD 20910

May 21-23
Custom Integrated Circuits Conference
Rochester, NY
Sponsor: Electron Devices Society of IEEE and the Rochester Section of IEEE
Contact: Dr. Aris Silzars, Tektronix, Inc., P.O. Box 500 MS 59-543, Beaverton, OR 97077

May 22-25
COMDEX Spring '84
Atlanta, GA
Contact: The Interface Group, 300 First Ave., Needham, MA 02194

JUNE

June 21-23
Great Southern Computer Show
Jacksonville, FL
Contact: Great Southern Computer Shows, P.O. Box 655, Jacksonville, FL 32201, 904-356-1044

What's the Best Feature of Davong's MultiLink Local Area Network System?

Just Ask Our Customers:

Faster: *"Remote operation with my Davong MultiLink™ is as fast as a stand alone IBM® PC XT."*

More Flexible: *"I found out you actually have to stop the whole Omninet™ network to increase disk capacity. You don't with Davong MultiLink.*

More Powerful: *"Our MultiLink supports five operating systems. 3Com only supports DOS."*

More Reliable: *". . . . the Arcnet™ technology Davong uses has been proven in over 5,000 installations."*

Less Expensive: *"Our 4-node MultiLink network saved us more than 40 % compared to a Nestar system."*

Davong MultiLink operates with IBM PC, XT or any IBM plug-compatible microcomputer like Compaq, Corona or Columbia. At an average work station cost of only about $700. Add hard disks or printers to any terminal on the network as you need them. Contact Davong or your local dealer for complete details. Let Davong MultiLink solve your local area network problems.

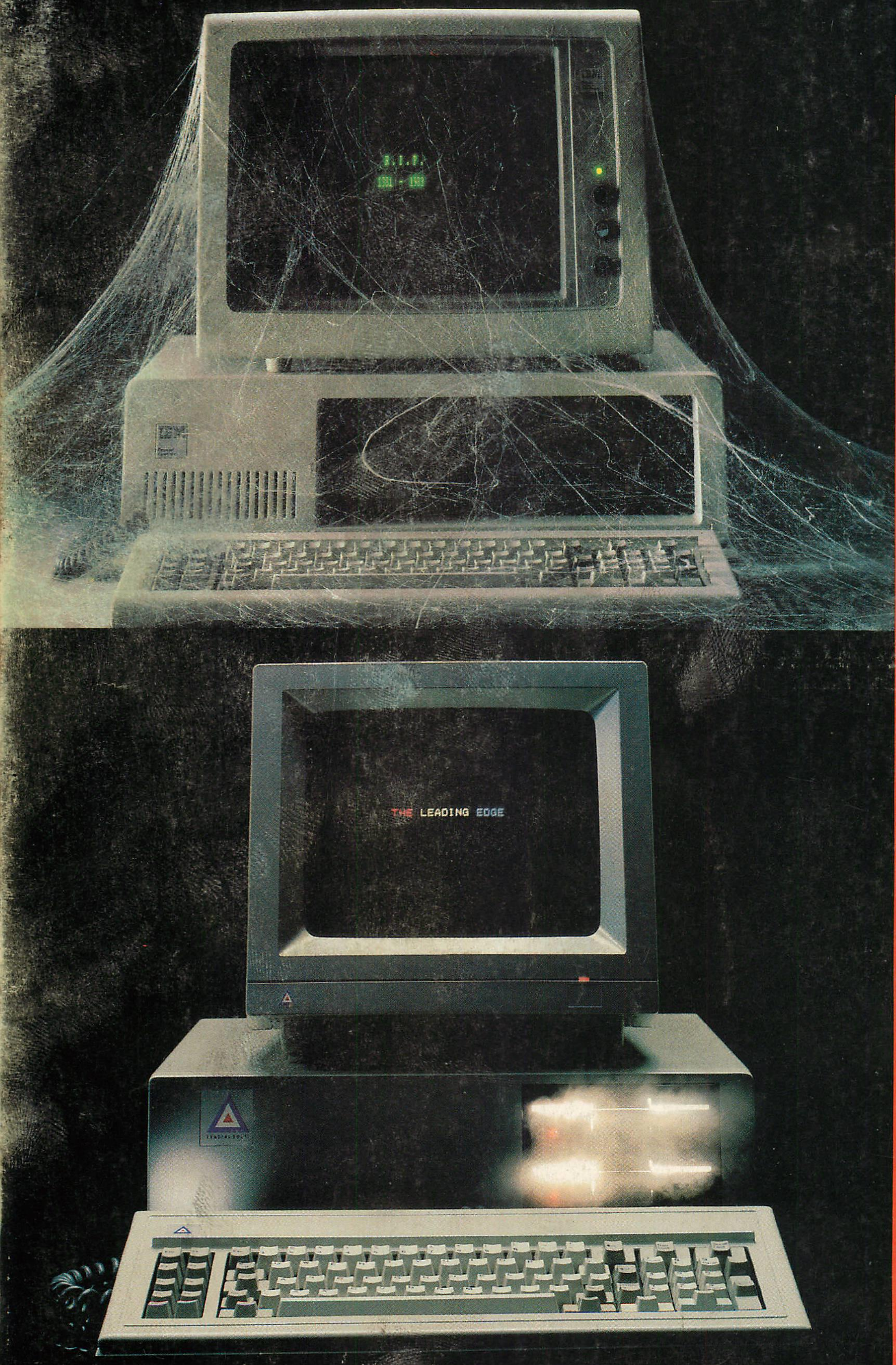
The Local Area Network Solution . . . Davong MultiLink

DAVONG

Davong Systems, Inc.
217 Humboldt Court
Sunnyvale, California 94089
Telephone: 408-734-4900

CIRCLE NO. 278 ON READER SERVICE CARD

THE DAY THE IBM® PC BECAME OBSOLETE.



It was a Monday in the autumn of '83:

The day they announced the Leading Edge® PC—a personal computer that's just plain better than the IBM® PC, at just about half the price.

The Leading Edge PC is faster (by more than 50%), more powerful, more flexible and more dependable (for example, our disk drives have a "mean time between failures" of 20,000 hours, versus an 8,000-hour MTBF for theirs). It's compatible with just about all the software and peripherals that the IBM is.

And unlike IBM's, ours comes complete with a high-resolution monitor, controller, seven expansion slots, serial port, parallel port, a time-of-day clock, double the standard memory (128K vs. 64K) plus hundreds of dollars worth of software to get you up and running immediately including MS™-DOS version 1.25, GW Basic, and Leading Edge Word Processing™ (the most powerful w.p. program ever created to run on an IBM-type personal computer). In short, the basic package comes to you complete and ready to work.

With IBM, on the other hand, you get charged extra for everything. Even for the PC DOS disk that makes it run (an extra $40) . . . and $170 just for the time of day (a calendar/clock that's standard with Leading Edge). In short, the basic package comes to you as a very expensive paperweight.

It's this simple: The Leading Edge Personal Computer is the first and only serious alternative to the IBM PC . . . and at only $2895 for the Leading Edge PC . . .

Get serious.

Now at 500 Leading Edge Dealers across the country. For the one nearest you, call toll free 1-800-343-6833. In Mass., call collect (617) 828-8150.

LEADING EDGE®

LEADING EDGE PRODUCTS, INC., SYSTEM SALES DIVISION, 225 TURNPIKE STREET, CANTON, MA 02021.

CIRCLE NO. 161 ON READER SERVICE CARD